Stream: Internet Engineering Task Force (IETF)
RFC: 9129
Category: Standards Track
Published: October 2022
ISSN: 2070-1721
Authors: D. Yeung, *Arrcus, Inc.*; Y. Qu, *Futurewei*; J. Zhang, *Juniper Networks*; I. Chen, *The MITRE Corporation*; A. Lindem, *Cisco Systems*

RFC 9129
YANG Data Model for the OSPF Protocol

Abstract

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

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9129.

Copyright Notice

Table of Contents

1. Introduction

YANG [RFC7950] is a data definition language used to define the contents of a conceptual datastore that allows networked devices to be managed using the Network Configuration Protocol (NETCONF) [RFC6241], RESTCONF [RFC8040], and other network management protocols. Furthermore, YANG data models can be used as the basis for implementation of other interfaces, such as Command-Line Interfaces (CLIs) and programmatic APIs.

This document defines a YANG data model that can be used to configure and manage OSPF. It is an augmentation to the core routing data model, which is defined in [RFC8349] and provides the basis for the development of data models for routing protocols. This document fully conforms to the Network Management Datastore Architecture (NMDA) [RFC8342]. The interface data model is defined in [RFC8343] and is used for referencing interfaces from the routing protocol. The data model for key chains [RFC8177] is used for OSPF authentication and provides both a reference to configured key chains and an enumeration of cryptographic algorithms.

Both OSPFv2 [RFC2328] and OSPFv3 [RFC5340] are supported. In addition to the core OSPF protocol, features described in other OSPF RFCs are also supported. These include demand circuits [RFC1793], Traffic Engineering (TE) [RFC3630], multiple address families [RFC5838], graceful restart [RFC3623] [RFC5187], the Not-So-Stubby Area (NSSA) option [RFC3101], and OSPFv2 or OSPFv3 as a Provider Edge to Customer Edge (PE-CE) protocol [RFC4577] [RFC6565]. These non-core features are optional in the OSPF data model.

1.1. Requirements Language

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Tree Diagrams

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

2. Design of Data Model

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

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

2.1. OSPF Operational State

The OSPF operational state is included in the same tree as OSPF configuration, consistent with the Network Management Datastore Architecture [RFC8342]. Consequently, only the "routing" container in the "ietf-routing" model [RFC8349] is augmented; the "routing-state" container is not augmented.

2.2. Overview

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

The OSPF YANG module augments the "/routing/control-plane-protocols/control-plane-protocol" path defined in the "ietf-routing" module. The "ietf-ospf" model defines a single instance of OSPF that may be instantiated as an OSPFv2 or OSPFv3 instance. Multiple instances are instantiated as multiple control-plane protocol instances.

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

The "ospf" container includes one OSPF protocol instance. The instance includes OSPF router-level configuration and operational state. Each OSPF instance maps to a control-plane protocol instance as defined in [RFC8349].

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

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

2.3. OSPFv2 and OSPFv3

The data model defined herein supports both OSPFv2 and OSPFv3.

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

2.4. Optional Features

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

This model defines the following optional features:

multi-topology: Support for Multi-Topology (MT) routing [RFC4915].

multi-area-adj: Support for OSPF multi-area adjacencies [RFC5185].

explicit-router-id: Support for the specification of an explicit per-instance Router ID.

demand-circuit: Support for OSPF demand circuits [RFC1793].

mtu-ignore: Support for the disabling of OSPF Database Description packet MTU mismatch checking as specified in Section 10.6 of [RFC2328].

lls: Support for OSPF Link-Local Signaling (LLS) [RFC5613].

prefix-suppression: Support for OSPF prefix advertisement suppression [RFC6860].

ttl-security: Support for OSPF Time to Live (TTL) security checking [RFC5082].

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

graceful-restart: Support for graceful OSPF restart [RFC3623] [RFC5187].

auto-cost: Support for OSPF interface cost calculations according to reference bandwidth [RFC2328].

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

max-lsa: Support for configuration of the maximum number of Link State Advertisements (LSAs) the OSPF instance will accept [RFC1765].

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

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

ospfv2-authentication-trailer: Support for the OSPFv2 authentication trailer [RFC5709] [RFC7474].

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

ospfv3-authentication-trailer: Support for the OSPFv3 authentication trailer [RFC7166].

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

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

node-tag: Support for node administrative tags for OSPF instances [RFC7777].

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

remote-lfa: Support for Remote LFAs (R-LFAs) [RFC7490].

stub-router: Support for OSPF stub router advertisements [RFC6987].

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

ietf-spf-delay: Support for the IETF Shortest Path First (SPF) delay algorithm [RFC8405].

bfd: Support for Bidirectional Forwarding Detection (BFD) to detect OSPF neighbor reachability [RFC5880] [RFC5881].

hybrid-interface: Support for OSPF hybrid broadcast and point-to-multipoint interfaces [RFC6845].

It is expected that vendors will support additional features through vendor-specific augmentations.

2.5. OSPF Router Configuration / Operational State

The "ospf" container is the top-level container in this data model. It represents an OSPF protocol instance and contains the router-level configuration and operational state. The operational state includes instance statistics, IETF SPF delay statistics, the AS-Scope Link State Database (LSDB), the local RIB, the SPF log, and the LSA log. ("AS" stands for "Autonomous System".)

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

```
      .
      +--rw address-family?       iana-rt-types:address-family
      +--rw enabled?              boolean
      +--rw explicit-router-id?   rt-types:router-id
      |                           {explicit-router-id}?
      +--rw preference
      |  +--rw (scope)?
      |     +--:(single-value)
      |     |  +--rw all?          uint8
      |     +--:(multi-values)
      |        +--rw (granularity)?
      |        |  +--:(detail)
      |        |  |  +--rw intra-area?   uint8
      |        |  |  +--rw inter-area?   uint8
      |        |  +--:(coarse)
      |        |     +--rw internal?     uint8
      |        +--rw external?     uint8
      +--rw nsr {nsr}?
      |  +--rw enabled?  boolean
      +--rw graceful-restart {graceful-restart}?
      |  +--rw enabled?                      boolean
      |  +--rw helper-enabled?               boolean
      |  +--rw restart-interval?             uint16
      |  +--rw helper-strict-lsa-checking?   boolean
      +--rw auto-cost {auto-cost}?
      |  +--rw enabled?               boolean
      |  +--rw reference-bandwidth?   uint32
      +--rw spf-control
      |  +--rw paths?            uint16 {max-ecmp}?
      |  +--rw ietf-spf-delay {ietf-spf-delay}?
      |     +--rw initial-delay?   uint32
      |     +--rw short-delay?     uint32
      |     +--rw long-delay?      uint32
      |     +--rw hold-down?       uint32
      |     +--rw time-to-learn?   uint32
      |     +--ro current-state?             enumeration
      |     +--ro remaining-time-to-learn?
      |                   rt-types:timer-value-milliseconds
      |     +--ro remaining-hold-down?
      |                   rt-types:timer-value-milliseconds
      |     +--ro last-event-received?       yang:timestamp
      |     +--ro next-spf-time?             yang:timestamp
      |     +--ro last-spf-time?             yang:timestamp
      +--rw database-control
      |  +--rw max-lsa?   uint32 {max-lsa}?
      +--rw stub-router {stub-router}?
      |  +--rw (trigger)?
      |     +--:(always)
      |        +--rw always!
      +--rw mpls
      |  +--rw te-rid {te-rid}?
      |  |  +--rw ipv4-router-id?   inet:ipv4-address
      |  |  +--rw ipv6-router-id?   inet:ipv6-address
      |  +--rw ldp
      |     +--rw igp-sync?   boolean {ldp-igp-sync}?
      +--rw fast-reroute {fast-reroute}?
      |  +--rw lfa {lfa}?
      +--rw node-tags {node-tag}?
```

```
        |  +--rw node-tag* [tag]
        |     +--rw tag      uint32
        +--ro router-id?          rt-types:router-id
        +--ro local-rib
        |  +--ro route* [prefix]
        |     +--ro prefix        inet:ip-prefix
        |     +--ro next-hops
        |     |  +--ro next-hop* []
        |     |     +--ro outgoing-interface?   if:interface-ref
        |     |     +--ro next-hop              inet:ip-address
        |     +--ro metric?       uint32
        |     +--ro route-type?   route-type
        |     +--ro route-tag?    uint32
        +--ro statistics
        |  +--ro discontinuity-time         yang:date-and-time
        |  +--ro originate-new-lsa-count?   yang:counter32
        |  +--ro rx-new-lsas-count?         yang:counter32
        |  +--ro as-scope-lsa-count?        yang:gauge32
        |  +--ro as-scope-lsa-chksum-sum?   uint32
        |  +--ro database
        |  |  +--ro as-scope-lsa-type*
        |  |     +--ro lsa-type?        uint16
        |  |     +--ro lsa-count?       yang:gauge32
        |  |     +--ro lsa-cksum-sum?   uint32
        |  +--ro protected-routes {fast-reroute}?
        |  |  +--ro address-family-stats*
        |  |         [address-family prefix alternate]
        |  |     +--ro address-family
        |  |         iana-rt-types:address-family
        |  |     +--ro prefix                  inet:ip-prefix
        |  |     +--ro alternate               inet:ip-address
        |  |     +--ro alternate-type?         enumeration
        |  |     +--ro best?                   boolean
        |  |     +--ro non-best-reason?        string
        |  |     +--ro protection-available?   bits
        |  |     +--ro alternate-metric-1?     uint32
        |  |     +--ro alternate-metric-2?     uint32
        |  |     +--ro alternate-metric-3?     uint32
        |  +--ro unprotected-routes {fast-reroute}?
        |  |  +--ro address-family-stats* [address-family prefix]
        |  |     +--ro address-family    iana-rt-types:address-family
        |  |     +--ro prefix            inet:ip-prefix
        |  +--ro protection-statistics* [frr-protection-method]
        |     +--ro frr-protection-method    string
        |     +--ro address-family-stats* [address-family]
        |        +--ro address-family
        |             iana-rt-types:address-family
        |        +--ro total-routes?           uint32
        |        +--ro unprotected-routes?     uint32
        |        +--ro protected-routes?       uint32
        |        +--ro linkprotected-routes?   uint32
        |        +--ro nodeprotected-routes?   uint32
        +--ro database
        |  +--ro as-scope-lsa-type* [lsa-type]
        |     +--ro as-scope-lsas
        |        +--ro as-scope-lsa* [lsa-id adv-router]
        |           +--ro lsa-id              union
        |           +--ro adv-router          inet:ipv4-address
```

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

2.6. OSPF Area Configuration / Operational State

The "area" container contains OSPF area configuration and the list of interface containers representing all the OSPF interfaces in the area. The area operational state includes area statistics and the area LSDB.

```
module: ietf-ospf
  augment /rt:routing/rt:control-plane-protocols/
           rt:control-plane-protocol:
    +--rw ospf
       .
       .
       +--rw areas
       |  +--rw area* [area-id]
       |     +--rw area-id                  area-id-type
       |     +--rw area-type?               identityref
       |     +--rw summary?                 boolean
```

```
        |     +--rw default-cost?            ospf-metric
        |     +--rw ranges
        |     |  +--rw range* [prefix]
        |     |     +--rw prefix       inet:ip-prefix
        |     |     +--rw advertise?   boolean
        |     |     +--rw cost?        ospf-metric
        |     +--rw topologies {ospf:multi-topology}?
        |     |  +--rw topology* [name]
        |     |     +--rw name  -> ../../../../../../../../
        |     |                    ../../../rt:ribs/rib/name
        |     |     +--rw summary?        boolean
        |     |     +--rw default-cost?   ospf-metric
        |     |     +--rw ranges
        |     |         +--rw range* [prefix]
        |     |            +--rw prefix       inet:ip-prefix
        |     |            +--rw advertise?   boolean
        |     |            +--rw cost?        ospf-metric
        |     +--ro statistics
        |     |  +--ro discontinuity-time           yang:date-and-time
        |     |  +--ro spf-runs-count?              yang:counter32
        |     |  +--ro abr-count?                   yang:gauge32
        |     |  +--ro asbr-count?                  yang:gauge32
        |     |  +--ro ar-nssa-translator-event-count?
        |     |                                     yang:counter32
        |     |  +--ro area-scope-lsa-count?        yang:gauge32
        |     |  +--ro area-scope-lsa-cksum-sum?    uint32
        |     |  +--ro database
        |     |     +--ro area-scope-lsa-type*
        |     |        +--ro lsa-type?        uint16
        |     |        +--ro lsa-count?       yang:gauge32
        |     |        +--ro lsa-cksum-sum?   uint32
        |     +--ro database
        |     |  +--ro area-scope-lsa-type* [lsa-type]
        |     |     +--ro lsa-type           uint16
        |     |     +--ro area-scope-lsas
        |     |        +--ro area-scope-lsa* [lsa-id adv-router]
        |     |           +--ro lsa-id               union
      .     .           .
      .     .           .
        |     |           +--ro (version)?
        |     |              +--:(ospfv2)
        |     |              |  +--ro ospfv2
        |     |              |     +--ro header
      .     .              .     .
      .     .              .     .
        |     |              |     +--ro body
        |     |              |        +--ro router
      .     .              .        .
      .     .              .        .
        |     |              |        +--ro network
      .     .              .        .
      .     .              .        .
        |     |              |        +--ro summary
      .     .              .        .
      .     .              .        .
        |     |              |        +--ro external
      .     .              .        .
      .     .              .        .
```

```
        |     |              |        +--ro opaque
        .     .              .        .
        .     .              .        .
        |     |              +--:(ospfv3)
        |     |                 +--ro ospfv3
        |     |                    +--ro header
        .     .                    .
        .     .                    .
        |     |                    +--ro body
        |     |                       +--ro router
        .     .                       .
        .     .                       .
        |     |                       +--ro network
        .     .                       .
        .     .                       .
        |     |                       +--ro inter-area-prefix
        .     .                       .
        .     .                       .
        |     |                       +--ro inter-area-router
        .     .                       .
        .     .                       .
        |     |                       +--ro as-external
        .     .                       .
        .     .                       .
        |     |                       +--ro nssa
        .     .                       .
        .     .                       .
        |     |                       +--ro link
        .     .                       .
        .     .                       .
        |     |                       +--ro intra-area-prefix
        .     .                       .
        .     .                       .
        |     |                       +--ro router-information
        .     .                       .
        .     .                       .
        |     +--rw virtual-links
        |     |  +--rw virtual-link* [transit-area-id router-id]
        |     |     +--rw transit-area-id       -> ../../../../
        |     |                                    area/area-id
        |     |     +--rw router-id             rt-types:router-id
        |     |     +--rw hello-interval?       uint16
        |     |     +--rw dead-interval?        uint32
        |     |     +--rw retransmit-interval?  uint16
        |     |     +--rw transmit-delay?       uint16
        |     |     +--rw lls?                  boolean {lls}?
        |     |     +--rw ttl-security {ttl-security}?
        |     |     |  +--rw enabled?  boolean
        |     |     |  +--rw hops?     uint8
        |     |     +--rw enabled?              boolean
        |     |     +--rw authentication
        |     |     |  +--rw (auth-type-selection)?
        |     |     |     +--:(ospfv2-auth)
        |     |     |     |  +--rw ospfv2-auth-trailer-rfc?
        |     |     |     |  |       ospfv2-auth-trailer-rfc-version
        |     |     |     |  |        {ospfv2-authentication-trailer}?
        |     |     |     |  +--rw (ospfv2-auth-specification)?
        |     |     |     |     +--:(auth-key-chain) {key-chain}?
```

```
        |     |     |     |     |  +--rw ospfv2-key-chain?
        |     |     |     |     |         key-chain:key-chain-ref
        |     |     |     |     +--:(auth-key-explicit)
        |     |     |     |        +--rw ospfv2-key-id?     uint32
        |     |     |     |        +--rw ospfv2-key?        string
        |     |     |     |        +--rw ospfv2-crypto-algorithm?
        |     |     |     |                 identityref
        |     |     |     +--:(ospfv3-auth-ipsec)
        |     |     |     |      {ospfv3-authentication-ipsec}?
        |     |     |     |  +--rw sa?                      string
        |     |     |     +--:(ospfv3-auth-trailer)
        |     |     |        |  {ospfv3-authentication-trailer}?
        |     |     |        +--rw (ospfv3-auth-specification)?
        |     |     |           +--:(auth-key-chain) {key-chain}?
        |     |     |           |  +--rw ospfv3-key-chain?
        |     |     |           |          key-chain:key-chain-ref
        |     |     |           +--:(auth-key-explicit)
        |     |     |              +--rw ospfv3-sa-id?        uint16
        |     |     |              +--rw ospfv3-key?          string
        |     |     |              +--rw ospfv3-crypto-algorithm?
        |     |     |                      identityref
        |     |     +--ro cost?              ospf-link-metric
        |     |     +--ro state?             if-state-type
        |     |     +--ro hello-timer?       rt-types:
        |     |     |                         rtimer-value-seconds16
        |     |     +--ro wait-timer?        rt-types:
        |     |     |                         rtimer-value-seconds16
        |     |     +--ro dr-router-id?      rt-types:router-id
        |     |     +--ro dr-ip-addr?        inet:ip-address
        |     |     +--ro bdr-router-id?     rt-types:router-id
        |     |     +--ro bdr-ip-addr?       inet:ip-address
        |     |     +--ro statistics
        |     |     |  +--ro discontinuity-time     yang:date-and-time
        |     |     |  +--ro if-event-count?        yang:counter32
        |     |     |  +--ro link-scope-lsa-count?  yang:gauge32
        |     |     |  +--ro link-scope-lsa-cksum-sum?
        |     |     |                               uint32
        |     |     |  +--ro database
        |     |     |     +--ro link-scope-lsa-type*
        |     |     |        +--ro lsa-type?        uint16
        |     |     |        +--ro lsa-count?       yang:gauge32
        |     |     |        +--ro lsa-cksum-sum?   int32
        |     |     +--ro neighbors
        |     |     |  +--ro neighbor* [neighbor-router-id]
        |     |     |     +--ro neighbor-router-id
        |     |     |                            rt-types:router-id
        |     |     |     +--ro address?         inet:ip-address
        |     |     |     +--ro dr-router-id?    rt-types:router-id
        |     |     |     +--ro dr-ip-addr?      inet:ip-address
        |     |     |     +--ro bdr-router-id?   rt-types:router-id
        |     |     |     +--ro bdr-ip-addr?     inet:ip-address
        |     |     |     +--ro state?           nbr-state-type
        |     |     |     +--ro dead-timer? rt-types:
        |     |     |     |                    rtimer-value-seconds16
        |     |     |     +--ro statistics
        |     |     |        +--ro discontinuity-time
        |     |     |                            yang:date-and-time
        |     |     |        +--ro nbr-event-count?
```

```
        |     |     |                            yang:counter32
        |     |     |       +--ro nbr-retrans-qlen?
        |     |     |                            yang:gauge32
        |     |     +--ro database
        |     |        +--ro link-scope-lsa-type* [lsa-type]
        |     |           +--ro lsa-type           uint16
        |     |           +--ro link-scope-lsas
        .     .
        .     .
        |     +--rw sham-links {pe-ce-protocol}?
        |     |  +--rw sham-link* [local-id remote-id]
        |     |     +--rw local-id               inet:ip-address
        |     |     +--rw remote-id              inet:ip-address
        |     |     +--rw hello-interval?        uint16
        |     |     +--rw dead-interval?         uint32
        |     |     +--rw retransmit-interval?   uint16
        |     |     +--rw transmit-delay?        uint16
        |     |     +--rw lls?                   boolean {lls}?
        |     |     +--rw ttl-security {ttl-security}?
        |     |     |  +--rw enabled?  boolean
        |     |     |  +--rw hops?     uint8
        |     |     +--rw enabled?            boolean
        |     |     +--rw authentication
        |     |     |  +--rw (auth-type-selection)?
        |     |     |     +--:(ospfv2-auth)
        |     |     |     |  +--rw ospfv2-auth-trailer-rfc?
        |     |     |     |  |       ospfv2-auth-trailer-rfc-version
        |     |     |     |  |        {ospfv2-authentication-trailer}?
        |     |     |     |  +--rw (ospfv2-auth-specification)?
        |     |     |     |     +--:(auth-key-chain) {key-chain}?
        |     |     |     |     |  +--rw ospfv2-key-chain?
        |     |     |     |     |         key-chain:key-chain-ref
        |     |     |     |     +--:(auth-key-explicit)
        |     |     |     |        +--rw ospfv2-key-id?     uint32
        |     |     |     |        +--rw ospfv2-key?        string
        |     |     |     |        +--rw ospfv2-crypto-algorithm?
        |     |     |     |                identityref
        |     |     |     +--:(ospfv3-auth-ipsec)
        |     |     |     |      {ospfv3-authentication-ipsec}?
        |     |     |     |  +--rw sa?                      string
        |     |     |     +--:(ospfv3-auth-trailer)
        |     |     |        |  {ospfv3-authentication-trailer}?
        |     |     |        +--rw (ospfv3-auth-specification)?
        |     |     |           +--:(auth-key-chain) {key-chain}?
        |     |     |           |  +--rw ospfv3-key-chain?
        |     |     |           |         key-chain:key-chain-ref
        |     |     |           +--:(auth-key-explicit)
        |     |     |              +--rw ospfv3-sa-id?        uint16
        |     |     |              +--rw ospfv3-key?          string
        |     |     |              +--rw ospfv3-crypto-algorithm?
        |     |     |                      identityref
        |     |     +--rw cost?              ospf-link-metric
        |     |     +--rw mtu-ignore?        boolean
        |     |                              {mtu-ignore}?
        |     |     +--rw prefix-suppression? boolean
        |     |                              {prefix-suppression}?
        |     |     +--ro state?             if-state-type
        |     |     +--ro hello-timer?      rt-types:
```

```
        |     |     |                           rtimer-value-seconds16
        |     |     +--ro wait-timer?        rt-types:
        |     |     |                           rtimer-value-seconds16
        |     |     +--ro dr-router-id?       rt-types:router-id
        |     |     +--ro dr-ip-addr?         inet:ip-address
        |     |     +--ro bdr-router-id?      rt-types:router-id
        |     |     +--ro bdr-ip-addr?        inet:ip-address
        |     |     +--ro statistics
        |     |     |  +--ro discontinuity-time     yang:date-and-time
        |     |     |  +--ro if-event-count?        yang:counter32
        |     |     |  +--ro link-scope-lsa-count?  yang:gauge32
        |     |     |  +--ro link-scope-lsa-cksum-sum?
        |     |     |                               uint32
        |     |     |  +--ro database
        |     |     |     +--ro link-scope-lsa-type*
        |     |     |        +--ro lsa-type?        uint16
        |     |     |        +--ro lsa-count?       yang:gauge32
        |     |     |        +--ro lsa-cksum-sum?   uint32
        |     |     +--ro neighbors
        |     |     |  +--ro neighbor* [neighbor-router-id]
        |     |     |     +--ro neighbor-router-id
        |     |     |                             rt-types:router-id
        |     |     |     +--ro address?          inet:ip-address
        |     |     |     +--ro dr-router-id?     rt-types:router-id
        |     |     |     +--ro dr-ip-addr?       inet:ip-address
        |     |     |     +--ro bdr-router-id?    rt-types:router-id
        |     |     |     +--ro bdr-ip-addr?      inet:ip-address
        |     |     |     +--ro state?            nbr-state-type
        |     |     |     +--ro cost?             ospf-link-metric
        |     |     |     +--ro dead-timer? rt-types:
        |     |     |     |                   rtimer-value-seconds16
        |     |     |     +--ro statistics
        |     |     |        +--ro discontinuity-time?
        |     |     |                             yang:date-and-time
        |     |     |        +--ro nbr-event-count?
        |     |     |                             yang:counter32
        |     |     |        +--ro nbr-retrans-qlen?
        |     |     |                             yang:gauge32
        |     |     +--ro database
        |     |        +--ro link-scope-lsa-type* [lsa-type]
        |     |           +--ro lsa-type           uint16
        |     |           +--ro link-scope-lsas
        .     .
        .     .
```

2.7. OSPF Interface Configuration / Operational State

The "interface" container contains OSPF interface configuration and operational state. The interface operational state includes the interface statistics, the list of neighbors, and the link-local LSDB.

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

```
        .
        .
        +--rw areas
        |  +--rw area* [area-id]
        |     .
        |     .
        |     +--rw interfaces
        |        +--rw interface* [name]
        |           +--rw name                    if:interface-ref
        |           +--rw interface-type?         enumeration
        |           +--rw passive?                boolean
        |           +--rw demand-circuit?         boolean
        |                                         {demand-circuit}?
        |           +--rw priority?               uint8
        |           +--rw multi-areas {multi-area-adj}?
        |           |  +--rw multi-area* [multi-area-id]
        |           |     +--rw multi-area-id      area-id-type
        |           |     +--rw cost?              ospf-link-metric
        |           +--rw static-neighbors
        |           |  +--rw neighbor* [identifier]
        |           |     +--rw identifier       inet:ip-address
        |           |     +--rw cost?            ospf-link-metric
        |           |     +--rw poll-interval?   uint16
        |           |     +--rw priority?        uint8
        |           +--rw node-flag?              boolean
        |                                         {node-flag}?
        |           +--rw bfd {bfd}?
        |           |  +--rw enabled?            boolean
        |           |  +--rw local-multiplier?   multiplier
        |           |  |      {client-base-cfg-parms}?
        |           |  +--rw (interval-config-type)?
        |           |  |      {client-base-cfg-parms}?
        |           |     +--:(tx-rx-intervals)
        |           |     |  +--rw desired-min-tx-interval?  uint32
        |           |     |  +--rw required-min-rx-interval? uint32
        |           |     +--:(single-interval)
        |           |     |   {single-minimum-interval}?
        |           |        +--rw min-interval?             uint32
        |           +--rw fast-reroute {fast-reroute}?
        |           |  +--rw lfa {lfa}?
        |           |     +--rw candidate-enabled?  boolean
        |           |     +--rw enabled?            boolean
        |           |     +--rw remote-lfa {remote-lfa}?
        |           |        +--rw enabled?  boolean
        |           +--rw hello-interval?         uint16
        |           +--rw dead-interval?          uint32
        |           +--rw retransmit-interval?    uint16
        |           +--rw transmit-delay?         uint16
        |           +--rw lls?                    boolean {lls}?
        |           +--rw ttl-security {ttl-security}?
        |           |  +--rw enabled?  boolean
        |           |  +--rw hops?     uint8
        |           +--rw enabled?                boolean
        |           +--rw authentication
        |           |  +--rw (auth-type-selection)?
        |           |     +--:(ospfv2-auth)
        |           |     |  +--rw ospfv2-auth-trailer-rfc?
        |           |     |  |       ospfv2-auth-trailer-rfc-version
```

```
|           |     |  |        {ospfv2-authentication-trailer}?
|           |     |  +--rw (ospfv2-auth-specification)?
|           |     |     +--:(auth-key-chain) {key-chain}?
|           |     |     |  +--rw ospfv2-key-chain?
|           |     |     |         key-chain:key-chain-ref
|           |     |     +--:(auth-key-explicit)
|           |     |        +--rw ospfv2-key-id?     uint32
|           |     |        +--rw ospfv2-key?        string
|           |     |        +--rw ospfv2-crypto-algorithm?
|           |     |                 identityref
|           |     +--:(ospfv3-auth-ipsec)
|           |     |      {ospfv3-authentication-ipsec}?
|           |     |  +--rw sa?                      string
|           |     +--:(ospfv3-auth-trailer)
|           |        |  {ospfv3-authentication-trailer}?
|           |        +--rw (ospfv3-auth-specification)?
|           |           +--:(auth-key-chain) {key-chain}?
|           |           |  +--rw ospfv3-key-chain?
|           |           |          key-chain:key-chain-ref
|           |           +--:(auth-key-explicit)
|           |              +--rw ospfv3-sa-id?        uint16
|           |              +--rw ospfv3-key?          string
|           |              +--rw ospfv3-crypto-algorithm?
|           |                      identityref
|           +--rw cost?              ospf-link-metric
|           +--rw mtu-ignore?        boolean
|           |                        {mtu-ignore}?
|           +--rw prefix-suppression? boolean
|           |                        {prefix-suppression}?
|           +--ro state?                if-state-type
|           +--ro hello-timer?       rt-types:
|           |                        rtimer-value-seconds16
|           +--ro wait-timer?        rt-types:
|           |                        rtimer-value-seconds16
|           +--ro dr-router-id?      rt-types:router-id
|           +--ro dr-ip-addr?        inet:ip-address
|           +--ro bdr-router-id?     rt-types:router-id
|           +--ro bdr-ip-addr?       inet:ip-address
|           +--ro statistics
|           |  +--ro discontinuity-time?    yang:date-and-time
|           |  +--ro if-event-count?        yang:counter32
|           |  +--ro link-scope-lsa-count?  yang:gauge32
|           |  +--ro link-scope-lsa-cksum-sum?
|           |                               uint32
|           |  +--ro database
|           |     +--ro link-scope-lsa-type*
|           |        +--ro lsa-type?        uint16
|           |        +--ro lsa-count?       yang:gauge32
|           |        +--ro lsa-cksum-sum?   int32
|           +--ro neighbors
|           |  +--ro neighbor* [neighbor-router-id]
|           |     +--ro neighbor-router-id
|           |                          rt-types:router-id
|           |     +--ro address?       inet:ip-address
|           |     +--ro dr-router-id?  rt-types:router-id
|           |     +--ro dr-ip-addr?    inet:ip-address
|           |     +--ro bdr-router-id? rt-types:router-id
|           |     +--ro bdr-ip-addr?   inet:ip-address
```

```
        |           |     +--ro state?          nbr-state-type
        |           |     +--ro dead-timer? rt-types:
        |           |     |                  rtimer-value-seconds16
        |           |     +--ro statistics
        |           |        +--ro discontinuity-time?
        |           |                         yang:date-and-time
        |           |        +--ro nbr-event-count?
        |           |                         yang:counter32
        |           |        +--ro nbr-retrans-qlen?
        |           |                         yang:gauge32
        |           +--ro database
        |           .  +--ro link-scope-lsa-type* [lsa-type]
        |           .     +--ro lsa-type           uint16
        |           .     +--ro link-scope-lsas
        .           .
        .           .
        |           +--rw topologies {ospf:multi-topology}?
        |           |  +--rw topology* [name]
        |           |     +--rw name  -> ../../../../../../../../
        |           |                    ../../../rt:ribs/rib/name
        |           |     +--rw cost? ospf-link-metric
        |           +--rw instance-id?           uint8
        .
        .
```

2.8. OSPF Notifications

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

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

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

```
     |  |  |     +--ro interface?   if:interface-ref
     |  |  +--:(virtual-link)
     |  |  |  +--ro virtual-link
     |  |  |     +--ro transit-area-id?      area-id-type
     |  |  |     +--ro neighbor-router-id?   rt-types:router-id
     |  |  +--:(sham-link)
     |  |     +--ro sham-link
     |  |        +--ro area-id?          area-id-type
     |  |        +--ro local-ip-addr?    inet:ip-address
     |  |        +--ro remote-ip-addr?   inet:ip-address
     |  +--ro neighbor-router-id?      rt-types:router-id
     |  +--ro neighbor-ip-addr?        yang:dotted-quad
     |  +--ro status?                  restart-helper-status-type
     |  +--ro age?                     rt-types:timer-value-seconds16
     |  +--ro exit-reason?             restart-exit-reason-type
     +---n if-rx-bad-packet
     |  +--ro routing-protocol-name?
     |  +     -> /rt:routing/control-plane-protocols/
     |  +         control-plane-protocol/name
     |  +--ro address-family?
     |  +     -> /rt:routing/control-plane-protocols/
     |  +         control-plane-protocol[rt:name=current()/../
     |  +         routing-protocol-name]/ospf/address-family
     |  +--ro (if-link-type-selection)?
     |  |  +--:(interface)
     |  |  |  +--ro interface
     |  |  |     +--ro interface?   if:interface-ref
     |  |  +--:(virtual-link)
     |  |  |  +--ro virtual-link
     |  |  |     +--ro transit-area-id?      area-id-type
     |  |  |     +--ro neighbor-router-id?   rt-types:router-id
     |  |  +--:(sham-link)
     |  |     +--ro sham-link
     |  |        +--ro area-id?          area-id-type
     |  |        +--ro local-ip-addr?    inet:ip-address
     |  |        +--ro remote-ip-addr?   inet:ip-address
     |  +--ro packet-source?           yang:dotted-quad
     |  +--ro packet-type?             packet-type
     +---n lsdb-approaching-overflow
     |  +--ro routing-protocol-name?
     |  +     -> /rt:routing/control-plane-protocols/
     |  +         control-plane-protocol/name
     |  +--ro address-family?
     |  +     -> /rt:routing/control-plane-protocols/
     |  +         control-plane-protocol[rt:name=current()/../
     |  +         routing-protocol-name]/ospf/address-family
     |  +--ro ext-lsdb-limit?          uint32
     +---n lsdb-overflow
     |  +--ro routing-protocol-name?
     |  +     -> /rt:routing/control-plane-protocols/
     |  +         control-plane-protocol/name
     |  +--ro address-family?
     |  +     -> /rt:routing/control-plane-protocols/
     |  +         control-plane-protocol[rt:name=current()/../
     |  +         routing-protocol-name]/ospf/address-family
     |  +--ro ext-lsdb-limit?          uint32
     +---n nssa-translator-status-change
     |  +--ro routing-protocol-name?
```

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

2.9. OSPF RPC Operations

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

clear-database: Resets the contents of a particular OSPF LSDB, forces neighbor adjacencies to the 'DOWN' state, and reoriginates self-originated LSAs.

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

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

3. OSPF YANG Module

The following RFCs are referenced in the "ietf-ospf" YANG module: [RFC0905], [RFC1765], [RFC1793], [RFC2328], [RFC3101], [RFC3623], [RFC3630], [RFC4552], [RFC4576], [RFC4577], [RFC4915], [RFC4973], [RFC5082], [RFC5185], [RFC5187], [RFC5250], [RFC5286], [RFC5309], [RFC5329], [RFC5340], [RFC5443], [RFC5613], [RFC5642], [RFC5709], [RFC5714], [RFC5838],

[RFC5880], [RFC5881], [RFC6565], [RFC6845], [RFC6860], [RFC6987], [RFC6991], [RFC7166], [RFC7474], [RFC7490], [RFC7684], [RFC7770], [RFC7777], [RFC7884], [RFC8177], [RFC8294], [RFC8343], [RFC8349], [RFC8405], [RFC8476], and [RFC9314].

```
<CODE BEGINS> file "ietf-ospf@2022-10-19.yang"

module ietf-ospf {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ospf";

  prefix ospf;

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

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

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

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

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

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

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

  import ietf-bfd-types {
    prefix bfd-types;
```

```
    reference
      "RFC 9314: YANG Data Model for Bidirectional Forwarding
       Detection (BFD)";
  }

  organization
    "IETF Link State Routing (lsr) Working Group";

  contact
    "WG Web:   <https://datatracker.ietf.org/wg/lsr/>
     WG List:  <mailto:lsr@ietf.org>

     Editor:   Derek Yeung
               <mailto:derek@arrcus.com>
     Author:   Acee Lindem
               <mailto:acee@cisco.com>
     Author:   Yingzhen Qu
               <mailto:yingzhen.qu@futurewei.com>
     Author:   Jeffrey Zhang
               <mailto:zzhang@juniper.net>
     Author:   Ing-Wher Chen
               <mailto:ingwherchen@mitre.org>";

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

     This YANG data model conforms to the Network Management
     Datastore Architecture (NMDA) as described in RFC 8342.

     The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
     NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED',
     'MAY', and 'OPTIONAL' in this document are to be interpreted as
     described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
     they appear in all capitals, as shown here.

     Copyright (c) 2022 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

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

     This version of this YANG module is part of RFC 9129; see the
     RFC itself for full legal notices.";

  revision 2022-10-19 {
    description
      "Initial revision.";
    reference
      "RFC 9129: YANG Data Model for the OSPF Protocol";
```

```
  }

  feature multi-topology {
    description
      "Support for Multi-Topology (MT) routing.";
    reference
      "RFC 4915: Multi-Topology (MT) Routing in OSPF";
  }

  feature multi-area-adj {
    description
      "Support for OSPF multi-area adjacencies as described in
       RFC 5185.";
    reference
      "RFC 5185: OSPF Multi-Area Adjacency";
  }

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

  feature demand-circuit {
    description
      "Support for OSPF demand circuits as defined in RFC 1793.";
    reference
      "RFC 1793: Extending OSPF to Support Demand Circuits";
  }

  feature mtu-ignore {
    description
      "Disable OSPF Database Description packet MTU
       mismatch checking as specified in the OSPFv2
       protocol specification (RFC 2328).  This mismatch checking
       also applies to OSPFv3 (RFC 5340).";
    reference
      "RFC 2328: OSPF Version 2, Section 10.6
       RFC 5340: OSPF for IPv6";
  }

  feature lls {
    description
      "OSPF link-local signaling (LLS) as defined in RFC 5613.";
    reference
      "RFC 5613: OSPF Link-Local Signaling";
  }

  feature prefix-suppression {
    description
      "OSPF prefix suppression support as described in RFC 6860.";
    reference
      "RFC 6860: Hiding Transit-Only Networks in OSPF";
  }

  feature ttl-security {
    description
      "Support for OSPF Time to Live (TTL) security checking.";
    reference
```

```
      "RFC 5082: The Generalized TTL Security Mechanism (GTSM)";
  }

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

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

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

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

  feature max-lsa {
    description
      "Sets the maximum number of Link State Advertisements (LSAs)
       the OSPF instance will accept.";
    reference
      "RFC 1765: OSPF Database Overflow";
  }

  feature te-rid {
    description
      "Support for configuration of the Traffic Engineering (TE)
       Router ID, i.e., the Router Address TLV as described in
       Section 2.4.1 of RFC 3630 or the Router IPv6 Address TLV
       as described in Section 3 of RFC 5329.";
    reference
      "RFC 3630: Traffic Engineering (TE) Extensions to
       OSPF Version 2, Section 2.4.1
       RFC 5329: Traffic Engineering Extensions to OSPF Version 3,
       Section 3";
  }

  feature ldp-igp-sync {
```

```
    description
      "LDP IGP synchronization.";
    reference
      "RFC 5443: LDP IGP Synchronization";
  }

  feature ospfv2-authentication-trailer {
    description
      "Support for the OSPFv2 authentication trailer.";
    reference
      "RFC 5709: OSPFv2 HMAC-SHA Cryptographic Authentication
       RFC 7474: Security Extension for OSPFv2 When
       Using Manual Key Management";
  }

  feature ospfv3-authentication-ipsec {
    description
      "Support for IPsec for OSPFv3 authentication.";
    reference
      "RFC 4552: Authentication/Confidentiality for OSPFv3";
  }

  feature ospfv3-authentication-trailer {
    description
      "Support for the OSPFv3 authentication trailer.";
    reference
      "RFC 7166: Supporting Authentication Trailer for OSPFv3";
  }

  feature fast-reroute {
    description
      "Support for IP Fast Reroute (IP-FRR).";
    reference
      "RFC 5714: IP Fast Reroute Framework";
  }

  feature key-chain {
    description
      "Support of key chains for authentication.";
    reference
      "RFC 8177: YANG Data Model for Key Chains";
  }

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

  feature node-tag {
    description
      "Support for node administrative tags for OSPF routing
       instances.";
    reference
      "RFC 7777: Advertising Node Administrative Tags in OSPF";
  }
```

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

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

  feature stub-router {
    description
      "Support for OSPF stub router advertisement as defined in
       RFC 6987.";
    reference
      "RFC 6987: OSPF Stub Router Advertisement";
  }

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

  feature ietf-spf-delay {
    description
      "Support for the IETF Shortest Path First (SPF) delay
       algorithm.";
    reference
      "RFC 8405: Shortest Path First (SPF) Back-Off Delay Algorithm
       for Link-State IGPs";
  }

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

  feature hybrid-interface {
    description
      "Support for the OSPF hybrid interface type.";
    reference
      "RFC 6845: OSPF Hybrid Broadcast and
```

```
       Point-to-Multipoint Interface Type";
  }

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

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

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

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

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

  identity stub-nssa-area {
    base area-type;
    description
      "OSPF stub area or Not-So-Stubby Area (NSSA).";
  }

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

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

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

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

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

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

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

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

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

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

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

  identity ospfv2-nssa-lsa {
    base ospfv2-external-lsa-type;
    description
      "OSPFv2 NSSA-LSA - Type 7.";
  }

  identity ospfv2-opaque-lsa-type {
    base ospfv2-lsa-type;
    description
```

```
      "OSPFv2 Opaque-LSA types.";
    reference
      "RFC 5250: The OSPF Opaque LSA Option";
  }

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

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

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

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

  identity ospfv3-lsa-type {
    base ospf-lsa-type;
    description
      "OSPFv3 LSA types.";
    reference
      "RFC 5340: OSPF for IPv6";
  }

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

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

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

  identity ospfv3-inter-area-prefix-lsa {
    base ospfv3-summary-lsa-type;
    description
```

```
      "OSPFv3 Inter-Area-Prefix-LSA - Type 0x2003.";
  }

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

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

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

  identity ospfv3-nssa-lsa {
    base ospfv3-external-lsa-type;
    description
      "OSPFv3 NSSA-LSA - Type 0x2007.";
  }

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

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

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

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

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

  identity lsa-refresh {
```

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

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

  identity lsa-purge {
    base lsa-log-reason;
    description
      "Identity used when an LSA is logged
       as a result of being purged.";
  }

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

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

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

  identity stub-router {
    base informational-capability;
    description
      "When set, the router is capable of acting as
       an OSPF stub router.";
    reference
      "RFC 6987: OSPF Stub Router Advertisement";
  }

  identity traffic-engineering {
    base informational-capability;
    description
```

```
      "When set, the router is capable of OSPF TE.";
    reference
      "RFC 3630: Traffic Engineering (TE) Extensions to
       OSPF Version 2
       RFC 5329: Traffic Engineering Extensions to OSPF Version 3";
  }

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

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

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

  identity vlink-end-bit {
    base router-lsa-bit;
    description
      "V-bit.  When set, the router is an endpoint of one or
       more virtual links.";
  }

  identity asbr-bit {
    base router-lsa-bit;
    description
      "E-bit.  When set, the router is an Autonomous System
       Boundary Router (ASBR).";
  }

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

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

```
  }

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

  identity af-bit {
    base ospfv3-lsa-option;
    description
      "AF-bit.  When set, the router supports OSPFv3 Address
       Families (AFs) as described in RFC 5838.";
    reference
      "RFC 5838: Support of Address Families in OSPFv3";
  }

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

  identity r-bit {
    base ospfv3-lsa-option;
    description
      "R-bit.  When set, the originator is an active router.";
  }

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

  identity e-bit {
    base ospfv3-lsa-option;
    description
      "E-bit.  This bit describes the way AS-External-LSAs
       are flooded.";
  }

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

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

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

```
  }

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

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

  identity dn-bit {
    base ospfv3-prefix-option;
    description
      "DN-bit.  When set, the Inter-Area-Prefix-LSA or
       AS-External-LSA prefix has been advertised as an
       L3VPN prefix.";
  }

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

  identity mt-bit {
    base ospfv2-lsa-option;
    description
      "MT-bit.  When set, the router supports multi-topology as
       described in RFC 4915.";
    reference
      "RFC 4915: Multi-Topology (MT) Routing in OSPF";
  }

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

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

  identity mc-bit {
    base ospfv2-lsa-option;
    description
      "MC-bit.  When set, the router supports
       Multicast Extensions to OSPF (MOSPF).";
```

```
  }

  identity v2-e-bit {
    base ospfv2-lsa-option;
    description
      "E-bit.  This bit describes the way AS-External-LSAs
       are flooded.";
  }

  identity o-bit {
    base ospfv2-lsa-option;
    description
      "O-bit.  When set, the router is opaque capable as described
       in RFC 5250.";
    reference
      "RFC 5250: The OSPF Opaque LSA Option";
  }

  identity v2-dn-bit {
    base ospfv2-lsa-option;
    description
      "DN-bit.  When a type 3, type 5, or type 7 LSA is sent from a
       PE to a CE, the DN-bit must be set.  See RFC 4576.";
    reference
      "RFC 4576: Using a Link State Advertisement (LSA) Options Bit
       to Prevent Looping in BGP/MPLS IP Virtual Private Networks
       (VPNs)";
  }

  identity ospfv2-extended-prefix-flag {
    description
      "Base identity for the Extended Prefix TLV flag.";
  }

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

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

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

```
  typedef ospf-link-metric {
    type uint16 {
      range "0 .. 65535";
    }
    description
      "OSPF link metric.  16-bit unsigned integer.";
  }

  typedef opaque-id {
    type uint32 {
      range "0 .. 16777215";
    }
    description
      "Opaque-LSA ID.  24-bit unsigned integer.";
  }

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

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

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

```
          "Interface is in the 'Down' state.";
      }
      enum loopback {
        value 2;
        description
          "Interface is in the 'Loopback' state.";
      }
      enum waiting {
        value 3;
        description
          "Interface is in the 'Waiting' state.";
      }
      enum point-to-point {
        value 4;
        description
          "Interface is in the 'Point-to-point' state.";
      }
      enum dr {
        value 5;
        description
          "Interface is in the 'DR' (Designated Router) state.";
      }
      enum bdr {
        value 6;
        description
          "Interface is in the 'Backup' (Backup Designated Router
           (BDR)) state.";
      }
      enum dr-other {
        value 7;
        description
          "Interface is in the 'DR Other' state.";
      }
    }
    description
      "OSPF interface state type.";
    reference
      "RFC 2328: OSPF Version 2";
  }

  typedef router-link-type {
    type enumeration {
      enum point-to-point-link {
        value 1;
        description
          "Point-to-point link to another router.";
      }
      enum transit-network-link {
        value 2;
        description
          "Link to a transit network, identified by the DR.";
      }
      enum stub-network-link {
        value 3;
        description
          "Link to a stub network, identified by the subnet.";
      }
      enum virtual-link {
```

```
        value 4;
        description
          "Virtual link across a transit area.";
      }
    }
    description
      "OSPF router link type.";
  }

  typedef nbr-state-type {
    type enumeration {
      enum down {
        value 1;
        description
          "Neighbor is in the 'Down' state.";
      }
      enum attempt {
        value 2;
        description
          "Neighbor is in the 'Attempt' state.";
      }
      enum init {
        value 3;
        description
          "Neighbor is in the 'Init' state.";
      }
      enum 2-way {
        value 4;
        description
          "Neighbor is in the '2-Way' state.";
      }
      enum exstart {
        value 5;
        description
          "Neighbor is in the 'ExStart' (exchange start) state.";
      }
      enum exchange {
        value 6;
        description
          "Neighbor is in the 'Exchange' state.";
      }
      enum loading {
        value 7;
        description
          "Neighbor is in the 'Loading' state.";
      }
      enum full {
        value 8;
        description
          "Neighbor is in the 'Full' state.";
      }
    }
    description
      "OSPF neighbor state type.";
    reference
      "RFC 2328: OSPF Version 2";
  }
```

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

  typedef restart-exit-reason-type {
    type enumeration {
      enum none {
        value 1;
        description
          "Restart not attempted.";
      }
      enum in-progress {
        value 2;
        description
          "Restart in progress.";
      }
      enum completed {
        value 3;
        description
          "Restart successfully completed.";
      }
      enum timed-out {
        value 4;
        description
          "Restart timed out.";
      }
      enum topology-changed {
        value 5;
        description
          "Restart aborted due to a topology change.";
      }
    }
    description
      "Describes the outcome of the last graceful restart attempt.
       The local router is being restarted or acting as a helper.";
  }

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

```
        value 2;
        description
          "OSPF Database Description packet.";
      }
      enum link-state-request {
        value 3;
        description
          "OSPF Link State Request packet.";
      }
      enum link-state-update {
        value 4;
        description
          "OSPF Link State Update packet.";
      }
      enum link-state-ack {
        value 5;
        description
          "OSPF Link State Acknowledgment packet.";
      }
    }
    description
      "OSPF packet type.";
  }

  typedef nssa-translator-state-type {
    type enumeration {
      enum enabled {
        value 1;
        description
          "NSSATranslatorState is 'enabled'.";
      }
      enum elected {
        value 2;
        description
          "NSSATranslatorState is 'elected'.";
      }
      enum disabled {
        value 3;
        description
          "NSSATranslatorState is 'disabled'.";
      }
    }
    description
      "OSPF NSSA translator state type.";
    reference
      "RFC 3101: The OSPF Not-So-Stubby Area (NSSA) Option";
  }

  typedef restart-status-type {
    type enumeration {
      enum not-restarting {
        value 1;
        description
          "The router is not restarting.";
      }
      enum planned-restart {
        value 2;
        description
```

```
          "The router is going through a planned restart.";
      }
      enum unplanned-restart {
        value 3;
        description
          "The router is going through an unplanned restart.";
      }
    }
    description
      "OSPF graceful restart status type.";
  }

  typedef fletcher-checksum16-type {
    type string {
      pattern '(0x)?[0-9a-fA-F]{4}';
    }
    description
      "Fletcher 16-bit checksum in hex-string format 0xXXXX.";
    reference
      "RFC 905: ISO Transport Protocol Specification ISO DP 8073";
  }

  typedef ospfv2-auth-trailer-rfc-version {
    type enumeration {
      enum rfc5709 {
        description
          "Support for the OSPF authentication trailer as
           described in RFC 5709.";
        reference
          "RFC 5709: OSPFv2 HMAC-SHA Cryptographic Authentication";
      }
      enum rfc7474 {
        description
          "Support for the OSPF authentication trailer as
           described in RFC 7474.";
        reference
          "RFC 7474: Security Extension for OSPFv2
           When Using Manual Key Management";
      }
    }
    description
      "OSPFv2 authentication trailer support.";
  }

  grouping tlv {
    description
      "Type-Length-Value (TLV).";
    leaf type {
      type uint16;
      description
        "TLV type.";
    }
    leaf length {
      type uint16;
      description
        "TLV length (octets).";
    }
    leaf value {
```

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

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

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

  grouping router-capabilities-tlv {
    description
      "Grouping for OSPF router capabilities TLV types.";
    reference
      "RFC 7770: Extensions to OSPF for Advertising Optional
       Router Capabilities";
    container router-informational-capabilities {
      leaf-list informational-capabilities {
        type identityref {
          base informational-capability;
        }
        description
          "List of informational capabilities.  This list will
           contain the identities for the informational
           capabilities supported by the router.";
      }
      description
        "OSPF Router Informational Flag definitions.";
    }
    list informational-capabilities-flags {
      leaf informational-flag {
        type uint32;
        description
          "Individual informational capability flag.";
```

```
      }
      description
        "List of informational capability flags.  This will
         return all the 32-bit informational flags, irrespective
         of whether or not they are known to the device.";
    }
    list functional-capabilities {
      leaf functional-flag {
        type uint32;
        description
          "Individual functional capability flag.";
      }
      description
        "List of functional capability flags.  This will
         return all the 32-bit functional flags, irrespective
         of whether or not they are known to the device.";
    }
  }

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

  grouping sbfd-discriminator-tlv {
    description
      "S-BFD Discriminator TLV.";
    reference
      "RFC 7884: OSPF Extensions to Advertise Seamless Bidirectional
       Forwarding Detection (S-BFD) Target Discriminators";
    list sbfd-discriminators {
      leaf sbfd-discriminator {
        type uint32;
        description
          "Individual S-BFD Discriminator.";
      }
      description
        "List of S-BFD Discriminators.";
    }
  }

  grouping maximum-sid-depth-tlv {
    description
      "Node MSD TLV (TLV for Maximum SID Depth).";
    reference
      "RFC 8476: Signaling Maximum SID Depth (MSD) Using OSPF";
    list msd-type {
      leaf msd-type {
        type uint8;
```

```
        description
          "Maximum SID Depth (MSD) type.";
      }
      leaf msd-value {
        type uint8;
        description
          "MSD value for the type.";
      }
      description
        "List of MSD tuples.";
    }
  }

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

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

```
  grouping ospfv2-lsa-body {
    description
      "OSPFv2 LSA body.";
    container router {
      when "derived-from-or-self(../../header/type, "
         + "'ospfv2-router-lsa')" {
        description
          "Only applies to Router-LSAs.";
      }
      description
        "Router-LSA.";
      uses ospf-router-lsa-bits;
      leaf num-of-links {
        type uint16;
        description
          "Number of links in the Router-LSA.";
      }
      container links {
        description
          "All router links.";
        list link {
          description
            "Router-LSA link.";
          uses ospfv2-router-link;
          container topologies {
            description
              "All topologies for the link.";
            list topology {
              description
                "Topology-specific information.";
              leaf mt-id {
                type uint8;
                description
                  "The MT-ID for the topology enabled on the link.";
              }
              leaf metric {
                type uint16;
                description
                  "Metric for the topology.";
              }
            }
          }
        }
      }
    }
    container network {
      when "derived-from-or-self(../../header/type, "
         + "'ospfv2-network-lsa')" {
        description
          "Only applies to Network-LSAs.";
      }
      description
        "Network-LSA.";
      leaf network-mask {
        type yang:dotted-quad;
        description
          "The IP address mask for the network.";
      }
```

```
        container attached-routers {
          description
            "All attached routers.";
          leaf-list attached-router {
            type inet:ipv4-address;
            description
              "List of the routers attached to the network.";
          }
        }
      }
      container summary {
        when "derived-from(../../header/type, "
           + "'ospfv2-summary-lsa-type')" {
          description
            "Only applies to summary LSAs.";
        }
        description
          "Summary LSA.";
        leaf network-mask {
          type inet:ipv4-address;
          description
            "The IP address mask for the network.";
        }
        container topologies {
          description
            "All topologies for the summary LSA.";
          list topology {
            description
              "Topology-specific information.";
            leaf mt-id {
              type uint8;
              description
                "The MT-ID for the topology enabled for the summary.";
            }
            leaf metric {
              type ospf-metric;
              description
                "Metric for the topology.";
            }
          }
        }
      }
      container external {
        when "derived-from(../../header/type, "
           + "'ospfv2-external-lsa-type')" {
          description
            "Only applies to AS-External-LSAs and NSSA-LSAs.";
        }
        description
          "External-LSA.";
        leaf network-mask {
          type inet:ipv4-address;
          description
            "The IP address mask for the network.";
        }
        container topologies {
          description
            "All topologies for the External-LSA.";
```

```
        list topology {
          description
            "Topology-specific information.";
          leaf mt-id {
            type uint8;
            description
              "The MT-ID for the topology enabled for the
               external or NSSA prefix.";
          }
          leaf flags {
            type bits {
              bit E {
                description
                  "When set, the metric specified is a Type 2
                   external metric.";
              }
            }
            description
              "Topology flags.";
          }
          leaf metric {
            type ospf-metric;
            description
              "Metric for the topology.";
          }
          leaf forwarding-address {
            type inet:ipv4-address;
            description
              "IPv4 Forwarding address.";
          }
          leaf external-route-tag {
            type uint32;
            description
              "Route tag for the topology.";
          }
        }
      }
    }
    container opaque {
      when "derived-from(../../header/type, "
         + "'ospfv2-opaque-lsa-type')" {
        description
          "Only applies to Opaque-LSAs.";
      }
      description
        "Opaque-LSA.";

      container ri-opaque {
        description
          "OSPF Router-Information-Opaque-LSA.";
        reference
          "RFC 7770: Extensions to OSPF for Advertising Optional
           Router Capabilities";

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

```
        }

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

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

        container sbfd-discriminator-tlv {
          description
            "OSPF S-BFD Discriminator TLV.";
          uses sbfd-discriminator-tlv;
        }

        container maximum-sid-depth-tlv {
          description
            "OSPF Node MSD TLV.";
          uses maximum-sid-depth-tlv;
        }
        uses unknown-tlvs;
      }

      container te-opaque {
        description
          "OSPFv2 TE Opaque-LSA.";
        reference
          "RFC 3630: Traffic Engineering (TE) Extensions to
           OSPF Version 2";

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

        container link-tlv {
          description
            "Describes a single link.  It is constructed
             from a set of sub-TLVs.";
          leaf link-type {
            type router-link-type;
            mandatory true;
            description
              "Link type.";
```

```
          }
          leaf link-id {
            type union {
              type inet:ipv4-address;
              type yang:dotted-quad;
            }
            mandatory true;
            description
              "Link ID.";
          }
          container local-if-ipv4-addrs {
            description
              "All local interface IPv4 addresses.";
            leaf-list local-if-ipv4-addr {
              type inet:ipv4-address;
              description
                "List of local interface IPv4 addresses.";
            }
          }
          container remote-if-ipv4-addrs {
            description
              "All remote interface IPv4 addresses.";
            leaf-list remote-if-ipv4-addr {
              type inet:ipv4-address;
              description
                "List of remote interface IPv4 addresses.";
            }
          }
          leaf te-metric {
            type uint32;
            description
              "TE metric.";
          }
          leaf max-bandwidth {
            type rt-types:bandwidth-ieee-float32;
            description
              "Maximum bandwidth.";
          }
          leaf max-reservable-bandwidth {
            type rt-types:bandwidth-ieee-float32;
            description
              "Maximum reservable bandwidth.";
          }
          container unreserved-bandwidths {
            description
              "All unreserved bandwidths.";
            list unreserved-bandwidth {
              leaf priority {
                type uint8 {
                  range "0 .. 7";
                }
                description
                  "Priority from 0 to 7.";
              }
              leaf unreserved-bandwidth {
                type rt-types:bandwidth-ieee-float32;
                description
                  "Unreserved bandwidth.";
```

```
            }
            description
              "List of unreserved bandwidths for different
               priorities.";
          }
        }
        leaf admin-group {
          type uint32;
          description
            "Administrative Group / Resource Class/Color.";
        }
        uses unknown-tlvs;
      }
    }

    container extended-prefix-opaque {
      description
        "All Extended Prefix TLVs in the LSA.";
      list extended-prefix-tlv {
        description
          "Extended Prefix TLV.";
        leaf route-type {
          type enumeration {
            enum unspecified {
              value 0;
              description
                "Unspecified.";
            }
            enum intra-area {
              value 1;
              description
                "OSPF intra-area route.";
            }
            enum inter-area {
              value 3;
              description
                "OSPF inter-area route.";
            }
            enum external {
              value 5;
              description
                "OSPF external route.";
            }
            enum nssa {
              value 7;
              description
                "OSPF NSSA external route.";
            }
          }
          description
            "Route type.";
        }
        container flags {
          leaf-list extended-prefix-flags {
            type identityref {
              base ospfv2-extended-prefix-flag;
            }
            description
```

```
                "List of Extended Prefix TLV flags.  This list will
                 contain identities for the prefix flags; these
                 identities are set in the extended prefix flags.";
              }
              description
                "Prefix flags.";
            }
            leaf prefix {
              type inet:ip-prefix;
              description
                "Address prefix.";
            }
            uses unknown-tlvs;
          }
        }

        container extended-link-opaque {
          description
            "All Extended Link TLVs in the LSA.";
          reference
            "RFC 7684: OSPFv2 Prefix/Link Attribute Advertisement";
          container extended-link-tlv {
            description
              "Extended Link TLV.";
            uses ospfv2-router-link;
            container maximum-sid-depth-tlv {
              description
                "OSPF Node MSD TLV.";
              uses maximum-sid-depth-tlv;
            }
            uses unknown-tlvs;
          }
        }
      }
    }

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

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

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

  grouping ospfv3-lsa-external {
    description
      "AS-External-LSA or NSSA-LSA.";
    leaf metric {
      type ospf-metric;
      description
        "AS-External-LSA or NSSA-LSA 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
        "AS-External-LSA or NSSA-LSA flags.";
    }

    leaf referenced-ls-type {
      type identityref {
        base ospfv3-lsa-type;
      }
      description
        "Referenced Link State (LS) Type.";
      reference
        "RFC 5340: OSPF for IPv6";
```

```
    }
    leaf unknown-referenced-ls-type {
      type uint16;
      description
        "Value for an unknown Referenced LS Type.";
    }

    uses ospfv3-lsa-prefix;

    leaf forwarding-address {
      type inet:ipv6-address;
      description
        "IPv6 Forwarding address.";
    }

    leaf external-route-tag {
      type uint32;
      description
        "Route tag.";
    }
    leaf referenced-link-state-id {
      type uint32;
      description
        "Referenced Link State ID.";
      reference
        "RFC 5340: OSPF for IPv6";
    }
  }

  grouping ospfv3-lsa-body {
    description
      "OSPFv3 LSA body.";
    container router {
      when "derived-from-or-self(../../header/type, "
         + "'ospfv3-router-lsa')" {
        description
          "Only applies to Router-LSAs.";
      }
      description
        "Router-LSA.";
      uses ospf-router-lsa-bits;
      uses ospfv3-lsa-options;

      container links {
        description
          "All router links.";
        list link {
          description
            "Router-LSA link.";
          leaf interface-id {
            type uint32;
            description
              "Interface ID for the link.";
          }
          leaf neighbor-interface-id {
            type uint32;
            description
              "Neighbor's Interface ID for the link.";
```

```
          }
          leaf neighbor-router-id {
            type rt-types:router-id;
            description
              "Neighbor's Router ID for the link.";
          }
          leaf type {
            type router-link-type;
            description
              "Link type: 1 - Point-to-Point Link
                          2 - Transit Network Link
                          3 - Reserved for OSPFv3 Links
                          4 - Virtual Link.";
          }
          leaf metric {
            type uint16;
            description
              "Link metric.";
          }
        }
      }
    }
    container network {
      when "derived-from-or-self(../../header/type, "
         + "'ospfv3-network-lsa')" {
        description
          "Only applies to Network-LSAs.";
      }
      description
        "Network-LSA.";

      uses ospfv3-lsa-options;

      container attached-routers {
        description
          "All attached routers.";
        leaf-list attached-router {
          type rt-types:router-id;
          description
            "List of the routers attached to the network.";
        }
      }
    }
    container inter-area-prefix {
      when "derived-from-or-self(../../header/type, "
         + "'ospfv3-inter-area-prefix-lsa')" {
        description
          "Only applies to Inter-Area-Prefix-LSAs.";
      }
      leaf metric {
        type ospf-metric;
        description
          "Inter-Area Prefix metric.";
      }
      uses ospfv3-lsa-prefix;
      description
        "Prefix-LSA.";
    }
```

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

        uses ospfv3-lsa-external;

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

        description
          "NSSA-LSA.";
      }
      container link {
        when "derived-from-or-self(../../header/type, "
           + "'ospfv3-link-lsa')" {
          description
            "Only applies to Link-LSAs.";
        }
        leaf rtr-priority {
          type uint8;
          description
            "Router priority for DR election.  A router with a
             higher priority will be preferred in the election.
             A value of 0 indicates that the router is not eligible
             to become the DR or BDR.";
        }
        uses ospfv3-lsa-options;
```

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

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

      container prefixes {
        description
          "All prefixes for the link.";
        list prefix {
          description
            "List of prefixes associated with the link.";
          uses ospfv3-lsa-prefix;
        }
      }
      description
        "Link-LSA.";
    }
    container intra-area-prefix {
      when "derived-from-or-self(../../header/type, "
         + "'ospfv3-intra-area-prefix-lsa')" {
        description
          "Only applies to Intra-Area-Prefix-LSAs.";
      }
      description
        "Intra-Area-Prefix-LSA.";

      leaf referenced-ls-type {
        type identityref {
          base ospfv3-lsa-type;
        }
        description
          "Referenced LS Type.";
      }
      leaf unknown-referenced-ls-type {
        type uint16;
        description
          "Value for an unknown Referenced LS Type.";
      }
      leaf referenced-link-state-id {
        type uint32;
        description
          "Referenced Link State ID.";
      }
      leaf referenced-adv-router {
        type rt-types:router-id;
        description
          "Referenced Advertising Router.";
        reference
          "RFC 5340: OSPF for IPv6";
```

```
      }

      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 uint16;
            description
              "Prefix metric.";
          }
        }
      }
    }
    container router-information {
      when "derived-from-or-self(../../header/type, "
         + "'ospfv3-router-information-lsa')" {
        description
          "Only applies to Router-Information-LSAs (RFC 7770).";
        reference
          "RFC 7770: Extensions to OSPF for Advertising Optional
           Router Capabilities";
      }
      container router-capabilities-tlv {
        description
          "Informational and functional router capabilities.";
        uses router-capabilities-tlv;
      }
      container node-tag-tlvs {
        description
          "All Node Admin Tag TLVs.";
        list node-tag-tlv {
          description
            "Node Admin Tag TLV.";
          uses node-tag-tlv;
        }
      }
      container dynamic-hostname-tlv {
        description
          "OSPF Dynamic Hostname TLV.";
        uses dynamic-hostname-tlv;
      }

      container sbfd-discriminator-tlv {
        description
          "OSPF S-BFD Discriminator TLV.";
        uses sbfd-discriminator-tlv;
      }

      description
```

```
        "Router-Information-LSA.";
      reference
        "RFC 7770: Extensions to OSPF for Advertising Optional
         Router Capabilities";
    }
  }

  grouping lsa-header {
    description
      "Common LSA for OSPFv2 and OSPFv3.";
    leaf age {
      type uint16;
      mandatory true;
      description
        "LSA age.";
    }
    leaf type {
      type identityref {
        base ospf-lsa-type;
      }
      mandatory true;
      description
        "LSA type.";
    }
    leaf adv-router {
      type rt-types:router-id;
      mandatory true;
      description
        "LSA advertising router.";
    }
    leaf seq-num {
      type uint32;
      mandatory true;
      description
        "LSA sequence number.";
    }
    leaf checksum {
      type fletcher-checksum16-type;
      mandatory true;
      description
        "LSA checksum.";
    }
    leaf length {
      type uint16;
      mandatory true;
      description
        "LSA length, including the header.";
    }
  }

  grouping ospfv2-lsa {
    description
      "OSPFv2 LSA.  LSAs are uniquely identified by
       the <LSA Type, Link State ID, Advertising Router>
       tuple, with the sequence number differentiating the
       LSA instances.";
    container header {
      must "(derived-from(type, "
```

```
          + "'ospfv2-opaque-lsa-type') and "
          + "opaque-id and opaque-type) or "
          + "(not(derived-from(type, "
          + "'ospfv2-opaque-lsa-type')) "
          + "and not(opaque-id) and not(opaque-type))" {
        description
          "The opaque-type and the opaque-id only apply to
           Opaque-LSAs.";
      }
      description
        "Decoded OSPFv2 LSA header data.";

      container lsa-options {
        leaf-list lsa-options {
          type identityref {
            base ospfv2-lsa-option;
          }
          description
            "List of LSA Options.  This list will contain the
             identities for the OSPFv2 LSA Options that are set.";
        }
        description
          "LSA Options.";
      }

      leaf lsa-id {
        type yang:dotted-quad;
        mandatory true;
        description
          "Link State ID.";
      }

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

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

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

  grouping ospfv3-lsa {
    description
      "Decoded OSPFv3 LSA.";
    container header {
      description
```

```
        "Decoded OSPFv3 LSA header data.";
      leaf lsa-id {
        type uint32;
        mandatory true;
        description
          "OSPFv3 LSA ID.";
      }
      uses lsa-header;
    }
    container body {
      description
        "Decoded OSPF LSA body data.";
      uses ospfv3-lsa-body;
    }
  }
  grouping lsa-common {
    description
      "Common fields for OSPF LSA representation.";
    leaf decode-completed {
      type boolean;
      description
        "The OSPF LSA body was successfully decoded, except for
         unknown TLVs.  Unknown LSA types and OSPFv2 unknown
         Opaque-LSA types are not decoded.  Additionally,
         malformed LSAs are generally not accepted and will
         not be in the Link State Database (LSDB).";
    }
    leaf raw-data {
      type yang:hex-string;
      description
        "The hexadecimal representation of the complete LSA as
         received or originated, in network byte order.";
    }
  }

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

  grouping lsa-key {
    description
      "OSPF LSA key.  The database key for each LSA of a given
```

```
        type in the LSDB.";
    leaf lsa-id {
      type union {
        type yang:dotted-quad;
        type uint32;
      }
      description
        "Link State ID.";
    }
    leaf adv-router {
      type rt-types:router-id;
      description
        "Advertising router.";
    }
  }

  grouping instance-stat {
    description
      "Per-instance statistics.";
    leaf discontinuity-time {
      type yang:date-and-time;
      description
        "The time of the most recent occasion at which any one or
         more of this OSPF instance's counters suffered a
         discontinuity.  If no such discontinuities have occurred
         since the OSPF instance was last reinitialized, then
         this node contains the time the OSPF instance was
         reinitialized, which normally occurs when it was
         created.";
    }
    leaf originate-new-lsa-count {
      type yang:counter32;
      description
        "The number of new LSAs originated.  Discontinuities in the
         value of this counter can occur when the OSPF instance is
         reinitialized.";
    }
    leaf rx-new-lsas-count {
      type yang:counter32;
      description
        "The number of new LSAs received.  Discontinuities in the
         value of this counter can occur when the OSPF instance is
         reinitialized.";
    }
    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 modulo 2^32 sum of the LSA checksums
         for AS-Scope LSAs.  The value should be treated as
         unsigned when comparing two sums of checksums.  While
         differing checksums indicate a different combination
         of LSAs, equivalent checksums don't guarantee that the
         LSAs are the same, given that multiple combinations of
```

```
          LSAs can result in the same checksum.";
      }
      container database {
        description
          "Container for per-AS-Scope LSA statistics.";
        list as-scope-lsa-type {
          description
            "List of AS-Scope LSA statistics.";
          leaf lsa-type {
            type uint16;
            description
              "AS-Scope LSA type.";
          }
          leaf lsa-count {
            type yang:gauge32;
            description
              "The number of LSAs of this LSA type.";
          }
          leaf lsa-cksum-sum {
            type uint32;
            description
              "The modulo 2^32 sum of the LSA checksums
               for LSAs of this type.  The value should be
               treated as unsigned when comparing two sums of
               checksums.  While differing checksums indicate a
               different combination of LSAs, equivalent checksums
               don't guarantee that the LSAs are the same, given that
               multiple combinations of LSAs can result in the same
               checksum.";
          }
        }
      }
      uses instance-fast-reroute-state;
    }

    grouping area-stat {
      description
        "Per-area statistics.";
      leaf discontinuity-time {
        type yang:date-and-time;
        description
          "The time of the most recent occasion at which any one or
           more of this OSPF area's counters suffered a
           discontinuity.  If no such discontinuities have occurred
           since the OSPF area was last reinitialized, then
           this node contains the time the OSPF area was
           reinitialized, which normally occurs when it was
           created.";
      }
      leaf spf-runs-count {
        type yang:counter32;
        description
          "The number of times the intra-area SPF has run.
           Discontinuities in the value of this counter can occur
           when the OSPF area is reinitialized.";
      }
      leaf abr-count {
        type yang:gauge32;
```

```
        description
          "The total number of Area Border Routers (ABRs)
           reachable within this area.";
      }
      leaf asbr-count {
        type yang:gauge32;
        description
          "The total number of AS Boundary Routers (ASBRs)
           reachable within this area.";
      }
      leaf ar-nssa-translator-event-count {
        type yang:counter32;
        description
          "The number of NSSA translator-state changes.
           Discontinuities in the value of this counter can occur
           when the OSPF area is reinitialized.";
      }
      leaf area-scope-lsa-count {
        type yang:gauge32;
        description
          "The number of area-scope LSAs in the area.";
      }
      leaf area-scope-lsa-cksum-sum {
        type uint32;
        description
          "The modulo 2^32 sum of the LSA checksums
           for area-scope LSAs.  The value should be treated as
           unsigned when comparing two sums of checksums.  While
           differing checksums indicate a different combination
           of LSAs, equivalent checksums don't guarantee that the
           LSAs are the same, given that multiple combinations of
           LSAs can result in the same checksum.";
      }
      container database {
        description
          "Container for area-scope LSA type statistics.";
        list area-scope-lsa-type {
          description
            "List of area-scope LSA statistics.";
          leaf lsa-type {
            type uint16;
            description
              "Area-scope LSA type.";
          }
          leaf lsa-count {
            type yang:gauge32;
            description
              "The number of LSAs of this LSA type.";
          }
          leaf lsa-cksum-sum {
            type uint32;
            description
              "The modulo 2^32 sum of the LSA checksums
               for LSAs of this type.  The value should be
               treated as unsigned when comparing two sums of
               checksums.  While differing checksums indicate a
               different combination of LSAs, equivalent checksums
               don't guarantee that the LSAs are the same, given that
```

```
            multiple combinations of LSAs can result in the same
            checksum.";
        }
      }
    }
  }

  grouping interface-stat {
    description
      "Per-interface statistics.";
    leaf discontinuity-time {
      type yang:date-and-time;
      description
        "The time of the most recent occasion at which any one or
         more of this OSPF interface's counters suffered a
         discontinuity.  If no such discontinuities have occurred
         since the OSPF interface was last reinitialized, then
         this node contains the time the OSPF interface was
         reinitialized, which normally occurs when it was
         created.";
    }
    leaf if-event-count {
      type yang:counter32;
      description
        "The number of times this interface has changed its
         state or an error has occurred.  Discontinuities in the
         value of this counter can occur when the OSPF interface
         is reinitialized.";
    }
    leaf link-scope-lsa-count {
      type yang:gauge32;
      description
        "The number of link-scope LSAs.";
    }
    leaf link-scope-lsa-cksum-sum {
      type uint32;
      description
        "The modulo 2^32 sum of the LSA checksums
         for link-scope LSAs.  The value should be treated as
         unsigned when comparing two sums of checksums.  While
         differing checksums indicate a different combination
         of LSAs, equivalent checksums don't guarantee that the
         LSAs are the same, given that multiple combinations of
         LSAs can result in the same checksum.";
    }
    container database {
      description
        "Container for link-scope LSA type statistics.";
      list link-scope-lsa-type {
        description
          "List of link-scope LSA statistics.";
        leaf lsa-type {
          type uint16;
          description
            "Link-scope LSA type.";
        }
        leaf lsa-count {
          type yang:gauge32;
```

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

  grouping neighbor-stat {
    description
      "Per-neighbor statistics.";
    leaf discontinuity-time {
      type yang:date-and-time;
      description
        "The time of the most recent occasion at which any one or
         more of this OSPF neighbor's counters suffered a
         discontinuity.  If no such discontinuities have occurred
         since the OSPF neighbor was last reinitialized, then
         this node contains the time the OSPF neighbor was
         reinitialized, which normally occurs when the neighbor
         is dynamically discovered and created.";
    }
    leaf nbr-event-count {
      type yang:counter32;
      description
        "The number of times this neighbor has changed
         state or an error has occurred.  Discontinuities in the
         value of this counter can occur when the OSPF neighbor
         is reinitialized.";
    }
    leaf nbr-retrans-qlen {
      type yang:gauge32;
      description
        "The current length of the retransmission queue.";
    }
  }

  grouping instance-fast-reroute-config {
    description
      "This group defines the global configuration of
       IP Fast Reroute (IP-FRR).";
    container fast-reroute {
      if-feature "fast-reroute";
      description
        "This container may be augmented with global
         parameters for IP-FRR.";
      container lfa {
```

```
        if-feature "lfa";
        description
          "This container may be augmented with
           global parameters for Loop-Free Alternates (LFAs).
           Container creation has no effect on LFA activation.";
      }
    }
  }

  grouping instance-fast-reroute-state {
    description
      "IP-FRR state data grouping.";

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

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

        leaf address-family {
          type iana-rt-types:address-family;
          description
            "Address family.";
        }
        leaf prefix {
          type inet:ip-prefix;
          description
            "Protected prefix.";
        }
        leaf alternate {
          type inet:ip-address;
          description
            "Alternate next hop for the prefix.";
        }
        leaf alternate-type {
          type enumeration {
            enum equal-cost {
              description
                "ECMP-based alternate.";
            }
            enum lfa {
              description
                "LFA-based alternate.";
            }
            enum remote-lfa {
              description
                "Remote-LFA-based alternate.";
            }
            enum tunnel {
              description
                "Tunnel-based alternate (like RSVP-TE or GRE).";
            }
            enum ti-lfa {
```

```
            description
              "An alternate based on Topology-Independent
               Loop-Free Alternate (TI-LFA).";
          }
          enum mrt {
            description
              "An alternate based on Maximally Redundant Trees
               (MRTs).";
          }
          enum other {
            description
              "Unknown alternate type.";
          }
        }
        description
          "Type of alternate.";
      }
      leaf best {
        type boolean;
        description
          "Indicates that this alternate is preferred.";
      }
      leaf non-best-reason {
        type string {
          length "1..255";
        }
        description
          "Information field used to describe why the alternate
           is not the best choice.";
      }
      leaf protection-available {
        type bits {
          bit node-protect {
            position 0;
            description
              "Node protection available.";
          }
          bit link-protect {
            position 1;
            description
              "Link protection available.";
          }
          bit srlg-protect {
            position 2;
            description
              "Shared Risk Link Group (SRLG) protection
               available.";
          }
          bit downstream-protect {
            position 3;
            description
              "Downstream protection available.";
          }
          bit other {
            position 4;
            description
              "Other protection available.";
          }
```

```
          }
          description
            "Protection provided by the alternate.";
        }
        leaf alternate-metric-1 {
          type uint32;
          description
            "Metric from the Point of Local Repair (PLR) to
             the destination through the alternate path.";
        }
        leaf alternate-metric-2 {
          type uint32;
          description
            "Metric from the PLR to the alternate node.";
        }
        leaf alternate-metric-3 {
          type uint32;
          description
            "Metric from the alternate node to the destination.";
        }
      }
    }

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

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

        leaf address-family {
          type iana-rt-types:address-family;
          description
            "Address family.";
        }
        leaf prefix {
          type inet:ip-prefix;
          description
            "Unprotected prefix.";
        }
      }
    }

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

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

```
      list address-family-stats {
        key "address-family";
        description
          "Per-AF protection statistics.";

        leaf address-family {
          type iana-rt-types:address-family;
          description
            "Address family.";
        }
        leaf total-routes {
          type uint32;
          description
            "Total prefixes.";
        }
        leaf unprotected-routes {
          type uint32;
          description
            "Total prefixes that are not protected.";
        }
        leaf protected-routes {
          type uint32;
          description
            "Total prefixes that are protected.";
        }
        leaf linkprotected-routes {
          type uint32;
          description
            "Total prefixes that are link protected.";
        }
        leaf nodeprotected-routes {
          type uint32;
          description
            "Total prefixes that are node protected.";
        }
      }
    }
  }

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

```
        }
        container remote-lfa {
          if-feature "remote-lfa";
          leaf enabled {
            type boolean;
            default "false";
            description
              "Activates a Remote LFA (R-LFA).";
          }
          description
            "R-LFA configuration.";
        }
        description
          "LFA configuration.";
      }
      description
        "Interface IP-FRR configuration.";
    }
  }

  grouping interface-physical-link-config {
    description
      "Interface cost configuration that only applies to
       physical interfaces (non-virtual) and sham links.";
    leaf cost {
      type ospf-link-metric;
      description
        "Interface's cost.";
    }
    leaf mtu-ignore {
      if-feature "mtu-ignore";
      type boolean;
      description
        "Enables/disables bypassing the MTU mismatch check in
         Database Description packets as specified in Section 10.6
         of RFC 2328.";
      reference
        "RFC 2328: OSPF Version 2, Section 10.6";
    }
    leaf prefix-suppression {
      if-feature "prefix-suppression";
      type boolean;
      description
        "Suppresses advertisement of the prefixes associated
         with the interface.";
    }
  }

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

    leaf hello-interval {
      type uint16;
      units "seconds";
      description
        "Interval between Hello packets (seconds).  It must
```

```
          be the same for all routers on the same network.
          Different networks, implementations, and deployments
          will use different Hello intervals.  A sample value
          for a LAN network would be 10 seconds.";
        reference
          "RFC 2328: OSPF Version 2, Appendix C.3";
      }

      leaf dead-interval {
        type uint16;
        units "seconds";
        must '../dead-interval > ../hello-interval' {
          error-message "The dead interval must be "
                      + "larger than the Hello interval";
          description
            "The value must be greater than 'hello-interval'.";
        }
        description
          "Interval after which a neighbor is declared down
           (seconds) if Hello packets are not received.  It is
           typically 3 or 4 times the 'hello-interval' period.
           A typical value for LAN networks is 40 seconds.";
        reference
          "RFC 2328: OSPF Version 2, Appendix C.3";
      }

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

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

      leaf lls {
        if-feature "lls";
        type boolean;
        description
          "Enables/disables link-local signaling (LLS) support.";
```

```
    }

    container ttl-security {
      if-feature "ttl-security";
      description
        "Time to Live (TTL) security checking.";
      leaf enabled {
        type boolean;
        description
          "Enables/disables TTL security checking.";
      }
      leaf hops {
        type uint8 {
          range "1..254";
        }
        default "1";
        description
          "Maximum number of hops that an OSPF packet may
           have traversed before reception.";
      }
    }
    leaf enabled {
      type boolean;
      default "true";
      description
        "Enables/disables the OSPF protocol on the interface.";
    }

    container authentication {
      description
        "Authentication configuration.";
      choice auth-type-selection {
        description
          "Options for OSPFv2/OSPFv3 authentication
           configuration.";
        case ospfv2-auth {
          when "derived-from-or-self(../../../../../../rt:type, "
             + "'ospfv2')" {
            description
              "Applied to OSPFv2 only.";
          }
          leaf ospfv2-auth-trailer-rfc {
            if-feature "ospfv2-authentication-trailer";
            type ospfv2-auth-trailer-rfc-version;
            description
              "Version of OSPFv2 authentication trailer support.
               See RFCs 5709 and 7474.";
            reference
              "RFC 5709: OSPFv2 HMAC-SHA Cryptographic Authentication
               RFC 7474: Security Extension for OSPFv2 When Using
               Manual Key Management";
          }
          choice ospfv2-auth-specification {
            description
              "Key chain or explicit key parameter specification.";
            case auth-key-chain {
              if-feature "key-chain";
              leaf ospfv2-key-chain {
```

```
                type key-chain:key-chain-ref;
                description
                  "Name of the key chain.";
              }
            }
            case auth-key-explicit {
              leaf ospfv2-key-id {
                type uint32;
                description
                  "Key identifier.";
              }
              leaf ospfv2-key {
                type string;
                description
                  "OSPFv2 authentication key.  The
                   length of the key may be dependent on the
                   cryptographic algorithm.";
              }
              leaf ospfv2-crypto-algorithm {
                type identityref {
                  base key-chain:crypto-algorithm;
                }
                description
                  "Cryptographic algorithm associated with the key.";
              }
            }
          }
        }
        case ospfv3-auth-ipsec {
          when "derived-from-or-self(../../../../../../rt:type, "
             + "'ospfv3')" {
            description
              "Applied to OSPFv3 only.";
          }
          if-feature "ospfv3-authentication-ipsec";
          leaf sa {
            type string;
            description
              "Name of the Security Association (SA).";
          }
        }
        case ospfv3-auth-trailer {
          when "derived-from-or-self(../../../../../../rt:type, "
             + "'ospfv3')" {
            description
              "Applied to OSPFv3 only.";
          }
          if-feature "ospfv3-authentication-trailer";
          choice ospfv3-auth-specification {
            description
              "Key chain or explicit key parameter specification.";
            case auth-key-chain {
              if-feature "key-chain";
              leaf ospfv3-key-chain {
                type key-chain:key-chain-ref;
                description
                  "Name of the key chain.";
              }
```

```
            }
            case auth-key-explicit {
              leaf ospfv3-sa-id {
                type uint16;
                description
                  "Security Association (SA) Identifier.";
              }
              leaf ospfv3-key {
                type string;
                description
                  "OSPFv3 authentication key.  The
                   length of the key may be dependent on the
                   cryptographic algorithm.";
              }
              leaf ospfv3-crypto-algorithm {
                type identityref {
                  base key-chain:crypto-algorithm;
                }
                description
                  "Cryptographic algorithm associated with the key.";
              }
            }
          }
        }
      }
    }
  }

  grouping interface-config {
    description
      "Configuration for normal OSPF interfaces (not virtual
       or sham interfaces).";

    leaf interface-type {
      type enumeration {
        enum broadcast {
          description
            "Specifies an OSPF broadcast multi-access network.";
        }
        enum non-broadcast {
          description
            "Specifies an OSPF Non-Broadcast Multi-Access
             (NBMA) network.";
        }
        enum point-to-multipoint {
          description
            "Specifies an OSPF point-to-multipoint network.";
        }
        enum point-to-point {
          description
            "Specifies an OSPF point-to-point network.";
        }
        enum hybrid {
          if-feature "hybrid-interface";
          description
            "Specifies an OSPF hybrid broadcast /
             point-to-multipoint network.";
        }
```

```
      }
      description
        "Interface type.";
    }

    leaf passive {
      type boolean;
      description
        "Enables/disables a passive interface.  A passive
         interface's prefix will be advertised, but no neighbor
         adjacencies will be formed on the interface.";
    }

    leaf demand-circuit {
      if-feature "demand-circuit";
      type boolean;
      description
        "Enables/disables a demand circuit.";
    }

    leaf priority {
      type uint8;
      description
        "Configures OSPF router priority.  In a multi-access
         network, this value is for Designated Router (DR) election.
         The priority is ignored on other interface types.  A router
         with a higher priority will be preferred in the election.
         A value of 0 indicates that the router is not eligible to
         become the DR or Backup DR (BDR).";
    }

    container multi-areas {
      if-feature "multi-area-adj";
      description
        "Container for multi-area configuration.";
      list multi-area {
        key "multi-area-id";
        description
          "Configures an OSPF multi-area adjacency.";
        leaf multi-area-id {
          type area-id-type;
          description
            "Multi-area adjacency area ID.";
        }
        leaf cost {
          type ospf-link-metric;
          description
            "Interface cost for a multi-area adjacency.";
        }
      }
    }

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

      list neighbor {
        key "identifier";
```

```
        description
          "Specifies a static OSPF neighbor.";

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

        leaf cost {
          type ospf-link-metric;
          description
            "Interface cost.  Different implementations have
             different default costs, with some defaulting to a
             cost inversely proportional to the interface speed.
             Others will default to 1, equating the cost to a
             hop count.";
        }
        leaf poll-interval {
          type uint16;
          units "seconds";
          description
            "Neighbor's poll interval (seconds) for sending OSPF
             Hello packets to discover the neighbor on NBMA
             networks.  This interval dictates the granularity for
             discovery of new neighbors.  A sample would be
             120 seconds (2 minutes) for a legacy Packet Data
             Network (PDN) X.25 network.";
          reference
            "RFC 2328: OSPF Version 2, Appendix C.5";
        }
        leaf priority {
          type uint8;
          description
            "Neighbor's priority for DR election.  A router with a
             higher priority will be preferred in the election.
             A value of 0 indicates that the router is not
             eligible to become the DR or BDR.";
        }
      }
    }

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

    container bfd {
      if-feature "bfd";
      description
        "BFD interface configuration.";
      uses bfd-types:client-cfg-parms;
      reference
```

```
        "RFC 5880: Bidirectional Forwarding Detection (BFD)
         RFC 5881: Bidirectional Forwarding Detection
         (BFD) for IPv4 and IPv6 (Single Hop)
         RFC 9314: YANG Data Model for Bidirectional Forwarding
         Detection (BFD)";
    }

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

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

    leaf address {
      type inet:ip-address;
      config false;
      description
        "Neighbor's address.";
    }
    leaf dr-router-id {
      type rt-types:router-id;
      config false;
      description
        "Neighbor's DR Router ID.";
    }

    leaf dr-ip-addr {
      type inet:ip-address;
      config false;
      description
        "Neighbor's DR IP address.";
    }

    leaf bdr-router-id {
      type rt-types:router-id;
      config false;
      description
        "Neighbor's BDR Router ID.";
    }

    leaf bdr-ip-addr {
      type inet:ip-address;
      config false;
      description
        "Neighbor's BDR IP address.";
    }
    leaf state {
      type nbr-state-type;
      config false;
      description
        "OSPF neighbor state.";
    }
    leaf cost {
      type ospf-link-metric;
      config false;
```

```
      description
        "Cost to reach the neighbor for point-to-multipoint
         and Hybrid networks.";
    }
    leaf dead-timer {
      type rt-types:timer-value-seconds16;
      config false;
      description
        "This timer tracks the remaining time before
         the neighbor is declared dead.";
    }
    container statistics {
      config false;
      description
        "Per-neighbor statistics.";
      uses neighbor-stat;
    }
  }

  grouping interface-common-state {
    description
      "OSPF interface common operational state.";
    reference
      "RFC 2328: OSPF Version 2, Section 9";

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

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

    leaf wait-timer {
      type rt-types:timer-value-seconds16;
      config false;
      description
        "This timer tracks the remaining time before
         the interface exits the 'Waiting' state.";
    }

    leaf dr-router-id {
      type rt-types:router-id;
      config false;
      description
        "DR Router ID.";
    }

    leaf dr-ip-addr {
      type inet:ip-address;
      config false;
```

```
        description
          "DR IP address.";
      }

      leaf bdr-router-id {
        type rt-types:router-id;
        config false;
        description
          "BDR Router ID.";
      }

      leaf bdr-ip-addr {
        type inet:ip-address;
        config false;
        description
          "BDR IP address.";
      }

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

      container neighbors {
        config false;
        description
          "All neighbors for the interface.";
        list neighbor {
          key "neighbor-router-id";
          description
            "List of interface OSPF neighbors.";
          leaf neighbor-router-id {
            type rt-types:router-id;
            description
              "Neighbor's Router ID.";
          }
          uses neighbor-state;
        }
      }
      container database {
        config false;
        description
          "Link-scope LSDB.";
        list link-scope-lsa-type {
          key "lsa-type";
          description
            "List of OSPF link-scope LSAs.";
          leaf lsa-type {
            type uint16;
            description
              "OSPF link-scope LSA type.";
          }
          container link-scope-lsas {
            description
              "All link-scope LSAs of this LSA type.";
            list link-scope-lsa {
```

```
            key "lsa-id adv-router";
            description
              "List of OSPF link-scope LSAs.";
            uses lsa-key;
            uses lsa {
              refine "version/ospfv2/ospfv2" {
                must "derived-from-or-self( "
                   + "../../../../../../../../../../"
                   + "rt:type, 'ospfv2')" {
                  description
                    "OSPFv2 LSA.";
                }
              }
              refine "version/ospfv3/ospfv3" {
                must "derived-from-or-self( "
                   + "../../../../../../../../../../"
                   + "rt:type, 'ospfv3')" {
                  description
                    "OSPFv3 LSA.";
                }
              }
            }
          }
        }
      }
    }
  }

  grouping interface-state {
    description
      "OSPF interface operational state.";
    reference
      "RFC 2328: OSPF Version 2, Section 9";

    uses interface-common-state;
  }

  grouping virtual-link-config {
    description
      "OSPF virtual link configuration state.";

    uses interface-common-config;
  }

  grouping virtual-link-state {
    description
      "OSPF virtual link operational state.";

    leaf cost {
      type ospf-link-metric;
      config false;
      description
        "Virtual link interface's cost.";
    }
    uses interface-common-state;
  }

  grouping sham-link-config {
```

```
    description
      "OSPF sham link configuration state.";

    uses interface-common-config;
    uses interface-physical-link-config;
  }

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

  grouping address-family-area-config {
    description
      "OSPF address-family-specific area configuration state.";

    container ranges {
      description
        "Container for summary ranges.";

      list range {
        key "prefix";
        description
          "Summarizes routes matching the address/mask.
           Applicable to Area Border Routers (ABRs) only.";
        leaf prefix {
          type inet:ip-prefix;
          description
            "IPv4 or IPv6 prefix.";
        }
        leaf advertise {
          type boolean;
          description
            "Advertise or hide.";
        }
        leaf cost {
          type ospf-metric;
          description
            "Advertised cost of a summary route.";
        }
      }
    }
  }

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

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

```
    }
    leaf default-cost {
      when "derived-from(../area-type,'stub-nssa-area')" {
        description
          "Cost for the LSA default route advertised into the
           stub area or NSSA.";
      }
      type ospf-metric;
      description
        "Sets the summary default route cost for a stub area
         or NSSA.";
    }
  }

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

    leaf area-type {
      type identityref {
        base area-type;
      }
      default "normal-area";
      description
        "Area type.";
    }

    uses area-common-config;
    uses address-family-area-config;
  }

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

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

    container database {
      config false;
      description
        "Area-scope LSDB.";
      list area-scope-lsa-type {
        key "lsa-type";
        description
          "List of OSPF area-scope LSAs.";
        leaf lsa-type {
          type uint16;
          description
            "OSPF area-scope LSA type.";
        }
        container area-scope-lsas {
          description
            "All area-scope LSAs.";
```

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

  grouping local-rib {
    description
      "Local RIB.  RIB for routes computed by the local
       OSPF routing instance.";
    container local-rib {
      config false;
      description
        "Local RIB.";
      list route {
        key "prefix";
        description
          "OSPF instance's Local Routes.";
        leaf prefix {
          type inet:ip-prefix;
          description
            "Destination prefix.";
        }
        container next-hops {
          description
            "Next hops for the route.";
          list next-hop {
            description
              "List of next hops for the route.";
            leaf outgoing-interface {
              type if:interface-ref;
              description
                "Name of the outgoing interface.";
            }
            leaf next-hop {
```

```
              type inet:ip-address;
              description
                "Address of the next hop.";
            }
          }
        }
        leaf metric {
          type uint32;
          description
            "Metric for this route.";
        }
        leaf route-type {
          type route-type;
          description
            "Route type for this route.";
        }
        leaf route-tag {
          type uint32;
          description
            "Route tag for this route.";
        }
      }
    }
  }

  grouping ietf-spf-delay {
    leaf initial-delay {
      type uint32;
      units "milliseconds";
      default "50";
      description
        "Delay used while in the 'QUIET' state (milliseconds).";
    }
    leaf short-delay {
      type uint32;
      units "milliseconds";
      default "200";
      description
        "Delay used while in the 'SHORT_WAIT' state (milliseconds).";
    }
    leaf long-delay {
      type uint32;
      units "milliseconds";
      default "5000";
      description
        "Delay used while in the 'LONG_WAIT' state (milliseconds).";
    }
    leaf hold-down {
      type uint32;
      units "milliseconds";
      default "10000";
      description
        "This timer value defines the period without any changes
         for the IGP to be considered stable (milliseconds).";
    }
    leaf time-to-learn {
      type uint32;
      units "milliseconds";
```

```
        default "500";
        description
          "Duration used to learn all the IGP events
           related to a single network event (milliseconds).";
      }
      leaf current-state {
        type enumeration {
          enum quiet {
            description
              "'QUIET' state.";
          }
          enum short-wait {
            description
              "'SHORT_WAIT' state.";
          }
          enum long-wait {
            description
              "'LONG_WAIT' state.";
          }
        }
        config false;
        description
          "Current SPF back-off algorithm state.";
      }
      leaf remaining-time-to-learn {
        type rt-types:timer-value-milliseconds;
        config false;
        description
          "Remaining time until the time-to-learn timer fires.";
      }
      leaf remaining-hold-down {
        type rt-types:timer-value-milliseconds;
        config false;
        description
          "Remaining time until the hold-down timer fires.";
      }
      leaf last-event-received {
        type yang:timestamp;
        config false;
        description
          "Time of the last SPF triggering event.";
      }
      leaf next-spf-time {
        type yang:timestamp;
        config false;
        description
          "Time when the next SPF has been scheduled.";
      }
      leaf last-spf-time {
        type yang:timestamp;
        config false;
        description
          "Time of the last SPF computation.";
      }
      description
        "Grouping for IETF SPF delay configuration and state.";
      reference
        "RFC 8405: Shortest Path First (SPF) Back-Off Delay Algorithm
```

```
       for Link-State IGPs";
  }

  grouping node-tag-config {
    description
      "OSPF node tag configuration state.";
    container node-tags {
      if-feature "node-tag";
      list node-tag {
        key "tag";
        leaf tag {
          type uint32;
          description
            "Node tag value.";
        }
        description
          "List of node tags.";
      }
      description
        "Container for node administrative tags.";
    }
  }

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

    leaf enabled {
      type boolean;
      default "true";
      description
        "Enables/disables the protocol.";
    }

    leaf explicit-router-id {
      if-feature "explicit-router-id";
      type rt-types:router-id;
      description
        "Defined in RFC 2328.  A 32-bit number
         that uniquely identifies the router.";
      reference
        "RFC 2328: OSPF Version 2";
    }

    container preference {
      description
        "Route preference configuration.  In many
         implementations, preference is referred to as
         administrative distance.";
      reference
        "RFC 8349: A YANG Data Model for Routing Management
         (NMDA Version)";
      choice scope {
        description
          "Options for expressing preference
           as single or multiple values.";
        case single-value {
          leaf all {
```

```
            type uint8;
            description
              "Preference for intra-area, inter-area, and
               external routes.";
          }
        }
        case multi-values {
          choice granularity {
            description
              "Options for expressing preference
               for intra-area and inter-area routes.";
            case detail {
              leaf intra-area {
                type uint8;
                description
                  "Preference for intra-area routes.";
              }
              leaf inter-area {
                type uint8;
                description
                  "Preference for inter-area routes.";
              }
            }
            case coarse {
              leaf internal {
                type uint8;
                description
                  "Preference for both intra-area and
                   inter-area routes.";
              }
            }
          }
          leaf external {
            type uint8;
            description
              "Preference for AS external and NSSA routes.";
          }
        }
      }
    }

    container nsr {
      if-feature "nsr";
      description
        "Non-Stop Routing (NSR) configuration state.";
      leaf enabled {
        type boolean;
        description
          "Enables/disables NSR.";
      }
    }

    container graceful-restart {
      if-feature "graceful-restart";
      description
        "Graceful restart configuration state.";
      reference
        "RFC 3623: Graceful OSPF Restart
```

```
          RFC 5187: OSPFv3 Graceful Restart";
        leaf enabled {
          type boolean;
          description
            "Enables/disables graceful restart as defined in RFC 3623
             for OSPFv2 and RFC 5187 for OSPFv3.";
        }
        leaf helper-enabled {
          type boolean;
          description
            "Enables graceful restart helper support for restarting
             routers (Section 3 of RFC 3623).";
          reference
            "RFC 3623: Graceful OSPF Restart, Section 3";
        }
        leaf restart-interval {
          type uint16 {
            range "1..1800";
          }
          units "seconds";
          default "120";
          description
            "Interval during which to attempt graceful restart prior
             to failing (seconds) (Appendix B.1 of RFC 3623).";
          reference
            "RFC 3623: Graceful OSPF Restart, Appendix B.1";
        }
        leaf helper-strict-lsa-checking {
          type boolean;
          description
            "Terminates graceful restart when an LSA topology change
             is detected (Appendix B.2 of RFC 3623).";
          reference
            "RFC 3623: Graceful OSPF Restart, Appendix B.2";
        }
      }

      container auto-cost {
        if-feature "auto-cost";
        description
          "Interface auto-cost configuration state.";
        leaf enabled {
          type boolean;
          description
            "Enables/disables interface auto-cost.";
        }
        leaf reference-bandwidth {
          when "../enabled = 'true'" {
            description
              "Only when auto-cost is enabled.";
          }
          type uint32 {
            range "1..4294967";
          }
          units "Mbits";
          description
            "Configures reference bandwidth used to automatically
             determine interface cost (Mbits).  The cost is the
```

```
            reference bandwidth divided by the interface speed,
            with 1 being the minimum cost.";
        }
      }

      container spf-control {
        leaf paths {
          if-feature "max-ecmp";
          type uint16 {
            range "1..65535";
          }
          description
            "Maximum number of Equal-Cost Multi-Path (ECMP) paths.";
        }
        container ietf-spf-delay {
          if-feature "ietf-spf-delay";
          uses ietf-spf-delay;
          description
            "IETF SPF delay algorithm configuration.";
        }
        description
          "SPF calculation control.";
      }

      container database-control {
        leaf max-lsa {
          if-feature "max-lsa";
          type uint32 {
            range "1..4294967294";
          }
          description
            "Maximum number of OSPF LSAs the router will accept.";
        }
        description
          "Database maintenance control.";
      }

      container stub-router {
        if-feature "stub-router";
        description
          "Sets the maximum metric configuration.";

        choice trigger {
          description
            "Specific triggers that will enable stub router state.";
          container always {
            presence "Enables unconditional stub router support";
            description
              "Unconditional stub router state (advertises
               transit links with 'MaxLinkMetric').";
            reference
              "RFC 6987: OSPF Stub Router Advertisement";
          }
        }
      }

      container mpls {
        description
```

```
        "OSPF MPLS configuration state.";
      container te-rid {
        if-feature "te-rid";
        description
          "Stable OSPF Router IP address used for TE.";
        leaf ipv4-router-id {
          type inet:ipv4-address;
          description
            "Explicitly configures a TE IPv4 Router ID.";
        }
        leaf ipv6-router-id {
          type inet:ipv6-address;
          description
            "Explicitly configures a TE IPv6 Router ID.";
        }
      }
      container ldp {
        description
          "OSPF MPLS LDP configuration state.";
        leaf igp-sync {
          if-feature "ldp-igp-sync";
          type boolean;
          description
            "Enables LDP IGP synchronization.";
        }
      }
    }
    uses instance-fast-reroute-config;
    uses node-tag-config;
  }

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

    leaf router-id {
      type rt-types:router-id;
      config false;
      description
        "Defined in RFC 2328.  A 32-bit number
         that uniquely identifies the router.";
      reference
        "RFC 2328: OSPF Version 2";
    }

    uses local-rib;

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

    container database {
      config false;
      description
        "AS-Scope LSDB.";
```

```
      list as-scope-lsa-type {
        key "lsa-type";
        description
          "List of OSPF AS-Scope LSAs.";
        leaf lsa-type {
          type uint16;
          description
            "OSPF AS-Scope LSA type.";
        }
        container as-scope-lsas {
          description
            "All AS-Scope LSAs of this LSA type.";
          list as-scope-lsa {
            key "lsa-id adv-router";
            description
              "List of OSPF AS-Scope LSAs.";
            uses lsa-key;
            uses lsa {
              refine "version/ospfv2/ospfv2" {
                must "derived-from-or-self( "
                   + "../../../../../../"
                   + "rt:type, 'ospfv2')" {
                  description
                    "OSPFv2 LSA.";
                }
              }
              refine "version/ospfv3/ospfv3" {
                must "derived-from-or-self( "
                   + "../../../../../../"
                   + "rt:type, 'ospfv3')" {
                  description
                    "OSPFv3 LSA.";
                }
              }
            }
          }
        }
      }
    }
    uses spf-log;
    uses lsa-log;
  }

  grouping multi-topology-area-common-config {
    description
      "OSPF multi-topology area common configuration state.";
    leaf summary {
      when "derived-from(../../../area-type, 'stub-nssa-area')" {
        description
          "Summary advertisement into the stub area or NSSA.";
      }
      type boolean;
      description
        "Enables/disables a summary advertisement into the
         topology in the stub area or NSSA.";
    }
    leaf default-cost {
      when "derived-from(../../../area-type, 'stub-nssa-area')" {
```

```
        description
          "Cost for the LSA default route advertised into the
           topology in the stub area or NSSA.";
      }
      type ospf-metric;
      description
        "Sets the summary default route cost for a
         stub area or NSSA.";
    }
  }

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

    uses multi-topology-area-common-config;
    uses address-family-area-config;
  }

  grouping multi-topology-state {
    description
      "OSPF multi-topology operational state.";

    uses local-rib;
  }

  grouping multi-topology-interface-config {
    description
      "OSPF multi-topology configuration state.";

    leaf cost {
      type ospf-link-metric;
      description
        "Interface cost for this topology.";
    }
  }

  grouping ospfv3-interface-config {
    description
      "OSPFv3 interface-specific configuration state.";

    leaf instance-id {
      type uint8;
      default "0";
      description
        "OSPFv3 instance ID.";
    }
  }

  grouping ospfv3-interface-state {
    description
      "OSPFv3 interface-specific operational state.";

    leaf interface-id {
      type uint32;
      config false;
      description
        "OSPFv3 interface ID.";
```

```
    }
  }

  grouping lsa-identifiers {
    description
      "The parameters that uniquely identify an LSA.";
    leaf area-id {
      type area-id-type;
      description
        "Area ID.";
    }
    leaf type {
      type uint16;
      description
        "LSA type.";
    }
    leaf lsa-id {
      type union {
        type inet:ipv4-address;
        type yang:dotted-quad;
      }
      description
        "Link State ID.";
    }
    leaf adv-router {
      type rt-types:router-id;
      description
        "LSA advertising router.";
    }
    leaf seq-num {
      type uint32;
      description
        "LSA sequence number.";
    }
  }

  grouping spf-log {
    description
      "Grouping for the SPF log.";
    container spf-log {
      config false;
      description
        "This container lists the SPF log entries.";
      list event {
        key "id";
        description
          "List of SPF log entries represented
           as a wrapping buffer in chronological
           order, with the oldest entry returned
           first.";
        leaf id {
          type uint32;
          description
            "Event identifier.  A purely internal value.";
        }
        leaf spf-type {
          type enumeration {
            enum full {
```

```
              description
                "The SPF computation was for a full SPF.";
            }
            enum intra {
              description
                "The SPF computation was only for intra-area
                 routes.";
            }
            enum inter {
              description
                "The SPF computation was only for inter-area
                 summary routes.";
            }
            enum external {
              description
                "The SPF computation was only for AS external
                 and NSSA routes.";
            }
          }
          description
            "The SPF computation type for the SPF log entry.";
        }
        leaf schedule-timestamp {
          type yang:timestamp;
          description
            "This is the timestamp when the computation was
             scheduled.";
        }
        leaf start-timestamp {
          type yang:timestamp;
          description
            "This is the timestamp when the computation was
             started.";
        }
        leaf end-timestamp {
          type yang:timestamp;
          description
            "This is the timestamp when the computation was
             completed.";
        }
        list trigger-lsa {
          description
            "The list of LSAs that triggered the computation.";
          uses lsa-identifiers;
        }
      }
    }
  }

  grouping lsa-log {
    description
      "Grouping for the LSA log.";
    container lsa-log {
      config false;
      description
        "This container lists the LSA log entries.
         Local LSA modifications are also included
         in the list.";
```

```
      list event {
        key "id";
        description
          "List of LSA log entries represented
           as a wrapping buffer in chronological order,
           with the oldest entry returned first.";
        leaf id {
          type uint32;
          description
            "Event identifier.  A purely internal value.";
        }
        container lsa {
          description
            "This container describes the LSA that was logged.";
          uses lsa-identifiers;
        }
        leaf received-timestamp {
          type yang:timestamp;
          description
            "This is the timestamp when the LSA was received.
             In the case of a local LSA update, the timestamp
             refers to the LSA origination time.";
        }
        leaf reason {
          type identityref {
            base lsa-log-reason;
          }
          description
            "Reason for the LSA log entry.";
        }
      }
    }
  }

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

    container ospf {
      description
        "OSPF protocol instance.";

      leaf address-family {
        when "derived-from-or-self(../../rt:type, 'ospfv3')" {
          description
            "Only applicable to OSPFv3.";
        }
        type iana-rt-types:address-family;
        description
          "Address family of the instance.";
      }
```

```
      uses instance-config;
      uses instance-state;

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

          uses area-config;
          uses area-state;

          container virtual-links {
            when "derived-from-or-self(../area-type, 'normal-area') "
               + "and ../area-id = '0.0.0.0'" {
              description
                "Virtual links must be in a backbone area.";
            }
            description
              "All virtual links.";
            list virtual-link {
              key "transit-area-id router-id";
              description
                "OSPF virtual link.";
              leaf transit-area-id {
                type leafref {
                  path "../../../../area/area-id";
                }
                must "derived-from-or-self("
                   + "../../../../area[area-id=current()]"
                   + "/area-type, 'normal-area') and "
                   + "../../../../area[area-id=current()]"
                   + "/area-id != '0.0.0.0'" {
                  error-message "The virtual link transit area must "
                              + "not be the backbone area.";
                  description
                    "The virtual link transit area must not be the
                     backbone area (0.0.0.0).";
                }
                description
                  "Virtual link transit area ID.";
              }
              leaf router-id {
                type rt-types:router-id;
                description
                  "Virtual link remote endpoint Router ID.";
              }

              uses virtual-link-config;
              uses virtual-link-state;
```

```
            }
          }
          container sham-links {
            if-feature "pe-ce-protocol";
            description
              "All sham links.";
            list sham-link {
              key "local-id remote-id";
              description
                "OSPF sham link.";
              leaf local-id {
                type inet:ip-address;
                description
                  "Address of the local sham link endpoint.";
              }
              leaf remote-id {
                type inet:ip-address;
                description
                  "Address of the remote sham link endpoint.";
              }
              uses sham-link-config;
              uses sham-link-state;
            }
          }
          container interfaces {
            description
              "All OSPF interfaces.";
            list interface {
              key "name";
              description
                "List of OSPF interfaces.";
              leaf name {
                type if:interface-ref;
                description
                  "Interface name reference.";
              }
              uses interface-config;
              uses interface-state;
            }
          }
        }
      }
    }
  }

  augment "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/ospf" {
    when "derived-from(../rt:type, 'ospf')" {
      description
        "This augmentation is only valid for OSPF
         (type 'ospfv2' or 'ospfv3').";
    }
    if-feature "multi-topology";
    description
      "OSPF multi-topology instance configuration
       state augmentation.";
    container topologies {
      description
```

```
        "All topologies.";
      list topology {
        key "name";
        description
          "OSPF topology.  The OSPF topology address family
           must coincide with the routing instance's
           address family.";
        leaf name {
          type leafref {
            path "../../../../../../rt:ribs/rt:rib/rt:name";
          }
          description
            "RIB name corresponding to the OSPF topology.";
        }

        uses multi-topology-state;
      }
    }
  }

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

        uses multi-topology-area-config;
      }
    }
  }

  augment "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/ospf/"
        + "areas/area/interfaces/interface" {
    when "derived-from-or-self(../../../../../rt:type, "
       + "'ospfv2')" {
```

```
      description
        "This augmentation is only valid for OSPFv2.";
    }
    if-feature "multi-topology";
    description
      "OSPF multi-topology interface configuration state
       augmentation.";
    container topologies {
      description
        "All topologies for the interface.";
      list topology {
        key "name";
        description
          "OSPF interface topology.";
        leaf name {
          type leafref {
            path "../../../../../../../../../../"
               + "rt:ribs/rt:rib/rt:name";
          }
          description
            "Single topology enabled on this interface.";
        }

        uses multi-topology-interface-config;
      }
    }
  }

  augment "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/ospf/"
        + "areas/area/interfaces/interface" {
    when "derived-from-or-self(../../../../../rt:type, "
       + "'ospfv3')" {
      description
        "This augmentation is only valid for OSPFv3.";
    }
    description
      "OSPFv3 interface-specific configuration state
       augmentation.";
    uses ospfv3-interface-config;
    uses ospfv3-interface-state;
  }

  grouping route-content {
    description
      "This grouping defines OSPF-specific route attributes.";
    leaf metric {
      type uint32;
      description
        "OSPF route metric.";
    }
    leaf tag {
      type uint32;
      default "0";
      description
        "OSPF route tag.";
    }
    leaf route-type {
```

```
      type route-type;
      description
        "OSPF route type.";
    }
  }

  augment "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route" {
    when "derived-from(rt:source-protocol, 'ospf')" {
      description
        "This augmentation is only valid for routes whose
         source protocol is OSPF.";
    }
    description
      "OSPF-specific route attributes.";
    uses route-content;
  }

  /*
   * RPCs
   */

  rpc clear-neighbor {
    description
      "This RPC request clears a particular set of OSPF neighbors.
       If the operation fails for an OSPF-internal reason, then
       'error-tag' and 'error-app-tag' should be set to values
       indicating the error.";
    input {
      leaf routing-protocol-name {
        type leafref {
          path "/rt:routing/rt:control-plane-protocols/"
             + "rt:control-plane-protocol/rt:name";
        }
        mandatory true;
        description
          "OSPF protocol instance for which information for neighbors
           is to be cleared.

           If the referenced OSPF instance doesn't exist, then
           this operation SHALL fail with an 'error-tag' setting of
           'data-missing' and an 'error-app-tag' setting of
           'routing-protocol-instance-not-found'.";
      }

      leaf interface {
        type if:interface-ref;
        description
          "Name of the OSPF interface for which neighbors are to
           be cleared.

           If the referenced OSPF interface doesn't exist, then
           this operation SHALL fail with an 'error-tag' setting
           of 'data-missing' and an 'error-app-tag' setting of
           'ospf-interface-not-found'.";
      }
    }
  }
```

```
  rpc clear-database {
    description
      "This RPC request clears a particular OSPF Link State
       Database.  Additionally, all neighbor adjacencies will
       be forced to the DOWN state and self-originated LSAs
       will be reoriginated.  If the operation fails for an
       OSPF-internal reason, then 'error-tag' and 'error-app-tag'
       should be set to values indicating the error.";
    input {
      leaf routing-protocol-name {
        type leafref {
          path "/rt:routing/rt:control-plane-protocols/"
             + "rt:control-plane-protocol/rt:name";
        }
        mandatory true;
        description
          "OSPF protocol instance whose LSDB is to be cleared.

           If the referenced OSPF instance doesn't exist, then
           this operation SHALL fail with an 'error-tag' setting of
           'data-missing' and an 'error-app-tag' setting of
           'routing-protocol-instance-not-found'.";
      }
    }
  }

  /*
   * Notifications
   */

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

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

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

```
  }

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

    choice if-link-type-selection {
      description
        "Options for link types.";
      container interface {
        description
          "Normal interface.";
        leaf interface {
          type if:interface-ref;
          description
            "Interface.";
        }
      }
      container virtual-link {
        description
          "Virtual link.";
        leaf transit-area-id {
          type area-id-type;
          description
            "Area ID.";
        }
        leaf neighbor-router-id {
          type rt-types:router-id;
          description
            "Neighbor's Router ID.";
        }
      }
      container sham-link {
        description
          "Sham link.";
        leaf area-id {
          type area-id-type;
          description
            "Area ID.";
        }
        leaf local-ip-addr {
          type inet:ip-address;
          description
            "Sham link's local address.";
        }
        leaf remote-ip-addr {
          type inet:ip-address;
          description
            "Sham link's remote address.";
        }
      }
    }
  }

  grouping notification-neighbor {
    description
      "This grouping provides the neighbor information
```

```
        for neighbor-specific notifications.";

    leaf neighbor-router-id {
      type rt-types:router-id;
      description
        "Neighbor's Router ID.";
    }

    leaf neighbor-ip-addr {
      type inet:ip-address;
      description
        "Neighbor's address.";
    }
  }

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

    leaf state {
      type if-state-type;
      description
        "Interface state.";
    }
    description
      "This notification is sent when an interface
       state change is detected.";
  }

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

    leaf packet-source {
      type inet:ip-address;
      description
        "Source address.";
    }

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

    leaf error {
      type enumeration {
        enum bad-version {
          description
            "Bad version.";
        }
        enum area-mismatch {
          description
            "Area mismatch.";
        }
        enum unknown-nbma-nbr {
          description
            "Unknown NBMA neighbor.";
```

```
        }
        enum unknown-virtual-nbr {
          description
            "Unknown virtual link neighbor.";
        }
        enum auth-type-mismatch {
          description
            "Authentication type mismatch.";
        }
        enum auth-failure {
          description
            "Authentication failure.";
        }
        enum net-mask-mismatch {
          description
            "Network mask mismatch.";
        }
        enum hello-interval-mismatch {
          description
            "Hello interval mismatch.";
        }
        enum dead-interval-mismatch {
          description
            "Dead interval mismatch.";
        }
        enum option-mismatch {
          description
            "Option mismatch.";
        }
        enum mtu-mismatch {
          description
            "MTU mismatch.";
        }
        enum duplicate-router-id {
          description
            "Duplicate Router ID.";
        }
        enum no-error {
          description
            "No error.";
        }
      }
      description
        "Error codes.";
    }
    description
      "This notification is sent when a packet is received indicating
       an interface configuration error on the sending OSPF router.";
  }

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

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

```
        "Neighbor state.";
    }

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

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

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

    leaf age {
      type rt-types:timer-value-seconds16;
      description
        "Remaining time in the current OSPF graceful restart
         interval when the router is acting as a restart
         helper for the neighbor.";
    }

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

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

    leaf packet-source {
      type inet:ip-address;
      description
        "Source address.";
    }

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

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

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

    leaf ext-lsdb-limit {
      type uint32;
      description
        "The maximum number of non-default AS-External-LSA
         entries that can be stored in the LSDB.";
    }

    description
      "This notification is sent when the number of LSAs
       in the router's LSDB has exceeded ninety percent of the
       AS-External-LSA limit ('ext-lsdb-limit').";
  }

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

    leaf ext-lsdb-limit {
      type uint32;
      description
        "The maximum number of non-default AS-External-LSA
         entries that can be stored in the LSDB.";
    }

    description
      "This notification is sent when the number of LSAs
       in the router's LSDB has exceeded the AS-External-LSA limit
       ('ext-lsdb-limit').";
  }

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

    leaf area-id {
      type area-id-type;
      description
        "Area ID.";
    }

    leaf status {
      type nssa-translator-state-type;
      description
        "NSSA translator status.";
    }

    description
      "This notification is sent when there is a change
       in the router's role in translating OSPF NSSA-LSAs
       to OSPF AS-External-LSAs.";
  }

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

    leaf status {
      type restart-status-type;
```

```
      description
        "Restart status.";
    }

    leaf restart-interval {
      type uint16 {
        range "1..1800";
      }
      units "seconds";
      default "120";
      description
        "Restart interval.";
    }

    leaf exit-reason {
      type restart-exit-reason-type;
      description
        "Restart exit reason.";
    }

    description
      "This notification is sent when the graceful restart
       state for the router has changed.";
  }
}

<CODE ENDS>
```

4. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

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

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

/ospf

/ospf/areas/

/ospf/areas/area[area-id]

/ospf/virtual-links/

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

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

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

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

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

Writable data nodes represent the configuration of each instance, area, virtual link, sham link, and interface, and they correspond to the schema nodes listed above.

For OSPF, the ability to modify OSPF configuration will allow the entire OSPF domain to be compromised, including peering with unauthorized routers to misroute traffic or mount a massive Denial-of-Service (DoS) attack. For example, adding OSPF on any unprotected interface could allow an OSPF adjacency to be formed with an unauthorized and malicious neighbor. Once an adjacency is formed, traffic could be hijacked. As a simpler example, a DoS attack could be mounted by changing the cost of an OSPF interface to be asymmetric such that a hard routing loop ensues. In general, unauthorized modification of most OSPF features will pose their own set of security risks. The Security Considerations sections in the respective reference RFCs should be consulted.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

/ospf/database

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

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

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

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

Exposure of the Link State Database (LSDB) will in turn expose the detailed topology of the network. There is a separate LSDB for each instance, area, virtual link, sham link, and interface. These correspond to the schema nodes listed above.

Exposure of the LSDB includes information beyond the scope of the OSPF router. This may be undesirable, since exposure may facilitate other attacks. Additionally, in the case of an area LSDB, the complete IP network topology and, if deployed, the TE topology of the OSPF area can be reconstructed. Network operators may consider their topologies to be sensitive confidential data.

For OSPF authentication, configuration is supported via the specification of key chains [RFC8177] or the direct specification of a key and an authentication algorithm. Hence, authentication configuration using the "auth-key-chain" case in the "ospfv2-auth-specification" or "ospfv3-auth-specification" container inherits the security considerations of [RFC8177]. This includes considerations with respect to the local storage and handling of authentication keys.

Additionally, local specification of OSPF authentication keys and the associated authentication algorithm is supported for legacy implementations that do not support key chains [RFC8177]. It is **RECOMMENDED** that implementations migrate to key chains because of (1) seamless support of key and algorithm rollover, (2) specification of a hexadecimal key, which affords more key entropy, and (3) encryption of keys using the Advanced Encryption Standard (AES) Key Wrap with Padding algorithm [RFC5649].

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

- The OSPF YANG module supports the "clear-neighbor" and "clear-database" RPCs. If access to either of these RPCs is compromised, temporary network outages can be employed to mount DoS attacks as a result.

The actual authentication key data (whether locally specified or part of a key chain) is sensitive and needs to be kept secret from unauthorized parties; compromise of the key data would allow an attacker to forge OSPF traffic that would be accepted as authentic, potentially compromising the entire OSPF domain.

5. IANA Considerations

This document registers a URI in the "IETF XML Registry" [RFC3688]. Following the format in [RFC3688], the following registration has been made:

URI: urn:ietf:params:xml:ns:yang:ietf-ospf
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

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

Name: ietf-ospf
Namespace: urn:ietf:params:xml:ns:yang:ietf-ospf
Prefix: ospf
Reference: RFC 9129

6. References

6.1. Normative References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[RFC9314] Jethanandani, M., Ed., Rahman, R., Ed., Zheng, L., Ed., Pallagatti, S., and G. Mirsky, "YANG Data Model for Bidirectional Forwarding Detection (BFD)", RFC 9314, DOI 10.17487/RFC9314, September 2022, <https://www.rfc-editor.org/info/rfc9314>.

6.2. Informative References

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

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

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

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

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

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

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

Acknowledgments

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

Thanks to Tom Petch for Last Call review and improvements to the organization of the document.

Thanks to Alvaro Retana for AD comments.

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

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

Contributors

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

Kiran Koushik Agrahara Sreenivasa
Verizon
500 W Dove Rd
Southlake, TX 76092
United States of America
Email: kk@employees.org

Authors' Addresses

Derek Yeung
Arrcus, Inc.
2077 Gateway Place, Suite 400
San Jose, CA 95110
United States of America
Email: derek@arrcus.com

Yingzhen Qu
Futurewei
2330 Central Expressway
Santa Clara, CA 95050
United States of America
Email: yingzhen.qu@futurewei.com

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

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

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC 27513
United States of America
Email: acee@cisco.com