

Internet
Internet-Draft
Intended status: Informational
Expires: April 25, 2014

D. Yeung
Y. Qu
A. Clemm
Cisco Systems
October 22, 2013

Yang Data Model for OSPF Protocol
draft-yeung-netmod-ospf-00

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 25, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Overview	2
1.1. Requirements Language	2
2. Design of Data Model	3
2.1. Overview	3
2.2. OSPFv2 and OSPFv3	4
2.3. Optional Features	4
2.4. Inheritance	4
2.5. OSPF Router Configuration	4
2.6. OSPF Address-Family Configuration	5
2.7. OSPF Area Configuration	6
2.8. OSPF Interface Configuration	7
3. OSPF Yang Module	9
4. Security Considerations	29
5. Acknowledgements	29
6. References	30
6.1. Normative References	30
6.2. Informative References	30

1. Overview

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

This document defines a YANG data model that can be used to configure and manage OSPF. Both OSPFv2 and OSPFv3 are supported. In addition to the core OSPF protocol, features described in different separate OSPF RFC are also supported. Those non-core features are made optional in the data model provided.

1.1. Requirements Language

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

2. Design of Data Model

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

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

2.1. Overview

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

```
module: ospf
  +--rw ospf-routers
    +--rw ospf-router [version name]
      .
      .
    +--rw ospf-afs
      +--rw ospf-af [vrf-name afi safi]
        .
        .
      +--rw ospf-areas
        .
        .
      +--rw ospf-area [area-id]
        .
        .
      +--rw ospf-interfaces
        .
        .
      +--rw ospf-interface [interface]
        .
        .
```

The container `ospf-routers` allows specification of one or more OSPF configuration entities called `ospf-router`. The `ospf-router` is intended to match to the vendor specific OSPF configuration construct which is identified by a local identifier 'name'. The field 'version' allows support for OSPFv2 and OSPFv3.

The container `ospf-afs` includes one or more OSPF protocol engines, each encapsulated in the `ospf-af` entity. Each `ospf-af` includes information for the routing domain it is running on based on the `[vrf-name afi safi]` specification. There is no default routing domain assumed by the data model. For example, to enable OSPF on the default IPv4 routing domain of the vendor, this model requires an explicit `ospf-af` entity with the specification like `["default" "ipv4" "unicast"]`. The `ospf-af` also contains OSPF router level configuration

The `ospf-areas/ospf-area` and `ospf-intefaces/ospf-interface` contains the OSPF configuration for the area and interface level respectively

2.2. OSPFv2 and OSPFv3

The defined data model supports both OSPFv2 and OSPFv3.

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

2.3. Optional Features

Optional features are features beyond the basic of OSPF configurations and it is up to a vendor to decide the support of a particular feature on a particular device.

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

2.4. Inheritance

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

Area related configurations specified in the `ospf-areas` container apply to all `ospf-area`.

Interface related configurations specified in `ospf-areas` container apply to all `ospf-interface` in all `ospf-area`'s.

Interface related configurations specified in `ospf-interfaces` container apply to all `ospf-interface` in the enclosing `ospf-area`.

2.5. OSPF Router Configuration

The container `ospf-routers` is the top level container in this data model. One or more `ospf-router` entity could be included in the `ospf-`

routers container. The main purpose of the ospf-router is to specify the OSPF version and the local name for the enclosed ospf-afs container which has the protocol configuration specifics.

```

module: ospf
  +--rw ospf-routers
    +--rw ospf-router [version name]
      +--rw version      uint8
      +--rw name         string
      .
      .

```

2.6. OSPF Address-Family Configuration

The container ospf-afs contains one or more ospf-af's each represents an OSPF protocol engine. Each ospf-af indicates the routing domain it is associated with based on [vrf-name afi safi] and contains the router level configurations.

```

module: ospf
  +--rw ospf-routers
    +--rw ospf-router [version name]
      ...
    +--rw ospf-afs
      +--rw ospf-af [vrf-name afi safi]
        +--rw vrf-name      string
        +--rw afi           enumeration
        +--rw safi          enumeration
        +--rw router-id
          | ...
        +--rw distance?     uint8
        +--rw default-originate
          | ...
        +--rw auto-cost
          | ...
        +--rw default-metric? uint32
        +--rw summary-prefix [prefix]
          | ...
        +--rw max-lsa
          | ...
        +--rw maximum
          | ...
        +--rw ignore
          | ...
        +--rw log
          | ...
        +--rw nsr

```

```

| ...
+--rw nsf
| ...
+--rw protocol
| ...
+--rw spf
| ...
+--rw snmp
| ...
+--rw redistribute
| ...
+--rw timers
| ...
+--rw mpls
| ...
+--rw fast-reroute
| ...

```

2.7. OSPF Area Configuration

The container `ospf-areas` contains one or more `ospf-area`'s each represents an area in the OSPF protocol. Each `ospf-area` contains configurations of that area and the `ospf-interfaces` container which includes all the OSPF interfaces active in the enclosing area.

The `ospf-areas` also contains area configuration that could be inherited to all `ospf-area`'s defined. Similarly, the `ospf-area` also contains interface configuration that could be inherited to all the `ospf-interface`'s defined.

```

module: ospf
  +--rw ospf-routers
    +--rw ospf-router [version name]
      ...
    +--rw ospf-afs
      +--rw ospf-af [vrf-name afi safi]
        ...
      +--rw ospf-areas
        +--rw loopback? enumeration
        +--rw network-type? enumeration
        +--rw passive? boolean
        +--rw mtu-ignore? boolean
        +--rw flood-reduction-enable? boolean
        +--rw demand-circuit? boolean
        +--rw demand-circuit-ignore-enable? boolean
        +--rw prefix-suppression? boolean

```

```

+--rw ttl-security
...
+--rw packet-size
...
+--rw authentication
...
+--rw database-filter
...
+--rw neighbor [ip-address]
...
+--rw bfd
...
+--rw cost?                               uint16
+--rw hello-interval?                     uint16
+--rw dead-interval?                      uint16
+--rw retransmit-interval?                uint16
+--rw transmit-delay?                    uint16
+--rw ospf-area [area-id]
  +--rw area-id                           uint32
  +--rw area-notation?                    enumeration
  +--rw (area-type)?
  ...
  +--rw default-cost?                      uint32
+--rw range [prefix]
...
+--rw virtual-link
...
+--rw sham-link
...
+--rw mpls-te-config
...

```

2.8. OSPF Interface Configuration

The container `ospf-areas` contains one or more `ospf-interface`'s each represents an interface in the OSPF protocol. Each `ospf-interface` contains configurations of that interface.

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

```

module: ospf
  +--rw ospf-routers
    +--rw ospf-router [version name]
      ...
    +--rw ospf-afs
      +--rw ospf-af [vrf-name afi safi]

```

```

...
+--rw ospf-areas
  ...
  +--rw ospf-interfaces
    +--rw network-type?          enumeration
    +--rw passive?              boolean
    +--rw mtu-ignore?           boolean
    +--rw flood-reduction-enable? boolean
    +--rw demand-circuit?      boolean
    +--rw demand-circuit-ignore-enable? boolean
    +--rw prefix-suppression?   boolean
    +--rw ttl-security
      ...
    +--rw packet-size
      ...
    +--rw authentication
      ...
    +--rw database-filter
      ...
    +--rw neighbor [ip-address]
      ...
    +--rw bfd
      ...
    +--rw cost?                  uint16
    +--rw hello-interval?       uint16
    +--rw dead-interval?        uint16
    +--rw retransmit-interval?   uint16
    +--rw transmit-delay?       uint16
    +--rw ospf-interface [interface]
      +--rw interface            if:interface-ref
      +--rw network-type?       enumeration
      +--rw passive?            boolean
      +--rw mtu-ignore?         boolean
      +--rw flood-reduction-enable? boolean
      +--rw demand-circuit?     boolean
      +--rw demand-circuit-ignore-enable? boolean
      +--rw prefix-suppression?   boolean
      +--rw ttl-security
        ...
      +--rw packet-size
        ...
      +--rw authentication
        ...
      +--rw database-filter
        ...
      +--rw neighbor [ip-address]
        ...
      +--rw bfd

```

```
    ...
    +--rw cost?                               uint16
    +--rw hello-interval?                     uint16
    +--rw dead-interval?                       uint16
    +--rw retransmit-interval?                 uint16
    +--rw transmit-delay?                      uint16
```

3. OSPF Yang Module

```
<CODE BEGINS>
module ospf {
  namespace "urn:cisco:params:xml:ns:yang:ospf";
  // replace with IANA namespace when assigned
  prefix ospf;

  import ietf-inet-types {
    prefix inet;
  }

  import ietf-interfaces {
    prefix if;
  }

  organization
    "Cisco Systems
     170 West Tasman Drive
     San Jose, CA 95134-1706
     USA";
  contact
    "Derek Yeung myeung@cisco.com
     Yingzhen Qu yiqu@cisco.com
     Alexander Clemm alex@cisco.com";

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

     Terms and Acronyms

     OSPF (ospf): Open Shortest Path First

     IP (ip): Internet Protocol
```

```
    IPv4 (ipv4):Internet Protocol Version 4
    IPv6 (ipv6): Internet Protocol Version 6
    MTU (mtu) Maximum Transmission Unit
";

revision 2013-10-20 {
  description
    "Initial revision.";
}

feature flood-reduction {
  description
    "OSPF Flood Reduction.";
}

feature demand-circuit-ignore {
  description
    "Ignore demand circuit auto-negotiation requests.";
}

feature security {
  description
    "Enable/Disable TTL security.";
}

feature packet-size {
  description
    "Customize size of OSPF packets up to MTU.";
}

feature nsf-cisco {
  description
    "Enable Cisco Non Stop Forwarding.";
}

feature loopback-config {
  description
    "OSPF loopback configuration.";
}

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

feature maximum-ecmp {
```

```
    description
      "Maximum number of ECMP paths.";
  }

  feature nsr {
    description
      "Enable NSR.";
  }

  feature spf {
    description
      "SPF configuration";
  }

  feature nssa-only {
    description
      "Limit summary to NSSA areas.";
  }

  feature nssa-ext-capability {
    description
      "Send domain specific capabilities into NSSA.";
  }

  grouping interface-internal-inherit-config {
    leaf cost {
      description
        "Interface cost.";
      type uint16 {
        range "1..65535";
      }
    }

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

    leaf dead-interval {
      description
        "Interval after which a neighbor is declared dead.";
      units seconds;
      type uint16 {
        range "1..65535";
      }
    }
  }
}
```

```
    }
    must "dead-interval > ../hello-interval" {
        error-app-tag dead-interval-invalid;
        error-message "The dead interval must be "
            + "larger than the hello interval";
    }
}

leaf retransmit-interval {
    description
        "Time between retransmitting lost link state
        advertisements.";
    units seconds;
    type uint16 {
        range "1..65535";
    }
}

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

grouping interface-inherit-config {
    leaf network-type {
        description
            "Network type.";
        type enumeration {
            enum "broadcast" {
                description
                    "Specify OSPF broadcast multi-access network.";
            }
            enum "non-broadcast" {
                description
                    "Specify OSPF NBMA network.";
            }
            enum "point-to-multipoint" {
                description
                    "Specify OSPF point-to-multipoint network.";
            }
            enum "point-to-point" {
                description
                    "Specify OSPF point-to-point network.";
            }
        }
    }
}
```

```
    }
  }
}

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

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

leaf flood-reduction-enable {
  if-feature flood-reduction;
  type boolean;
}

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

leaf demand-circuit-ignore-enable {
  if-feature demand-circuit-ignore;
  type boolean;
}

leaf prefix-suppression {
  description
    "Suppress advertisement of the prefixes.";
  type boolean;
}

container ttl-security {
  if-feature security;
  leaf enable {
    type boolean;
  }
  leaf hops {
    type uint8 {
      range "1..254";
    }
  }
}
```

```
    }  
  
    container packet-size {  
      if-feature packet-size;  
      description  
        "Customize size of OSPF packets up to MTU.";  
      leaf enable {  
        type boolean;  
      }  
      leaf size {  
        type uint16 {  
          range "576..10000";  
        }  
      }  
    }  
  }  
}
```

```
container authentication {  
  when "../.../version = 2";  
  description "Configure authentication.";  
  leaf enable {  
    type boolean;  
  }  
  choice authentication-type-selection {  
    case clear-text {  
      description  
        "Use clear-text authentication.";  
      leaf key {  
        type string {  
          length "1..8";  
        }  
      }  
    }  
    case message-digest {  
      description  
        "Use message-digest authentication.";  
      list message-digest-key {  
        key key-id;  
        leaf key-id {  
          type uint8 {  
            range "1..255";  
          }  
        }  
        leaf algorithm {  
          type enumeration {  
            enum "md5";  
          }  
        }  
        leaf key {
```

```
        type string {
            length "1..8";
        }
    }
}

container database-filter {
    description
        "Filter OSPF LSA during synchronization and flooding";
    leaf enable {
        type boolean;
    }
    leaf lsa-type {
        type enumeration {
            enum "all";
        }
    }
    leaf direction {
        type enumeration {
            enum "out";
        }
    }
}

list neighbor {
    description
        "Specify a neighbor router.";
    key ip-address;
    leaf ip-address {
        type inet:ip-address;
    }
    leaf cost {
        type uint16 {
            range "1..65535";
        }
    }
    leaf poll-interval {
        units seconds;
        type uint16 {
            range "1..65535";
        }
    }
    leaf priority {
        type uint8 {
            range "1..255";
        }
    }
}
```

```
    }
  }
}

container bfd {
  leaf enable {
    description
      "enable bfd.";
    type boolean;
  }

  leaf minimum-interval {
    description
      "bfd hello interval";
    type uint16 {
      range "15..30000";
    }
  }
  leaf multiplier {
    description
      "detect multiplier";
    type uint8 {
      range "2..50";
    }
  }
}

uses interface-internal-inherit-config;

} // grouping interface-inherit-config

grouping area-inherit-config {
  leaf loopback {
    if-feature loopback-config;
    description
      "OSPF loopback configuration.";
    type enumeration {
      enum "stub";
      enum "host";
    }
  }
}
} // grouping area-inherit-config

container ospf-routers {
  list ospf-router {
    description
      "This is a top-level container for the OSPF router.";
  }
}
```

```
key "version name";
leaf version {
  description
    "OSPF version.";
  mandatory true;
  type uint8 {
    range "2..3";
  }
}

leaf name {
  description
    "Name, combined with
    ospf-routers/ospf_router/ospf-afs/ospf-af,
    identify an OSPF protocol instance.";
  mandatory true;
  type string;
}

container ospf-afs {
  list ospf-af {
    key "vrf-name afi safi";
    leaf vrf-name {
      type string;
    }
    leaf afi {
      type enumeration {
        enum "ipv4" {
          description
            "IPv4";
        }
        enum "ipv6" {
          description
            "IPv6";
        }
      }
    }
    leaf safi {
      type enumeration {
        enum "unicast" {
          description
            "unicast";
        }
      }
    }
  }

  container router-id {
    description
```

```
        "Specify the router-id for this OSPF process.";
    leaf enable {
        type boolean;
    }
    choice config-type {
        case static {
            leaf ip-address {
                type boolean;
            }
        }
        case auto-config {
            leaf enable-auto-config {
                type boolean;
            }
        }
    }
}

leaf distance {
    description
        "Define an administrative distance.";
    type uint8 {
        range "1..255";
    }
}

container default-originate {
    description
        "Control distribution of default route information.";
    leaf enable {
        type boolean;
        default "false";
    }
}

container auto-cost {
    description
        "Calculate OSPF interface cost according to
        bandwidth.";
    leaf enable {
        type boolean;
    }
    leaf reference-bandwidth {
        type uint32 {
            range "1..4294967";
        }
    }
}
}
```

```
leaf default-metric {
  description
    "Calculate OSPF interface cost according to
    bandwidth.";
  type uint32 {
    range "1..16777214";
  }
}

list summary-prefix {
  description
    "Configure IP address summaries";
  key "prefix";
  leaf prefix {
    type inet:ip-prefix;
  }
  leaf advertise {
    type boolean;
  }
  leaf tag {
    type uint32 {
      range "0..4294967295";
    }
  }
  leaf nssa-only-enable {
    if-feature nssa-only;
    type boolean;
  }
}

container max-lsa {
  if-feature max-lsa;
  description
    "Maximum number of LSAs OSPF process will
    receive.";
  leaf max {
    description
      "Maximum number of non self-generated LSAs
      this process can receive.";
    type uint32 {
      range "1..4294967294";
    }
  }
  leaf threshold {
    description
      "Threshold value (%) at which to generate
      a warning msg.";
    type uint8 {
```

```
        range "1..100";
    }
}
leaf warning-only {
    description
        "Only give warning message when limit is
        exceeded.";
    type boolean;
}
leaf ignore-time {
    description
        "Number of minutes during which all adjacencies
        are suppressed.";
    type uint16 {
        range "1..17895";
    }
}
leaf ignore-count {
    description
        "Count on how many times adjacencies can be
        suppressed.";
    type uint16 {
        range "1..65534";
    }
}
leaf reset-time {
    description
        "number of minutes after which ignore-count is
        reset to zero.";
    type uint16 {
        range "2..35791";
    }
}
}

container maximum {
    if-feature maximum-ecmp;
    description
        "Set OSPF limits.";
    leaf paths {
        description
            "Maximum number of ECMP paths.";
        type uint16 {
            range "1..32";
        }
    }
}
leaf interfaces {
    description
```

```
        "Maximum number of interfaces.";
    type uint16 {
        range "1..1024";
    }
}
container redistributed-prefixes {
    description
        "Maximum number of prefixes redistributed.";
    leaf max {
        type uint32 {
            range "1..4294967295";
        }
    }
    leaf threshold {
        description
            "Threshold value (%) at which to generate
            a warning msg.";
        type uint8 {
            range "1..100";
        }
    }
    leaf warning-only {
        description
            "Only give warning message when limit is
            exceeded.";
        type boolean;
    }
}
}

container ignore {
    description
        "Do not complain about specific event.";
    leaf-list lsa {
        type enumeration {
            enum "mospf";
        }
    }
}

container log {
    description
        "Log ospf info.";
    leaf adjacency-changes {
        type boolean;
    }
}
}
```

```
    container nsr {
      if-feature nsr;
      description
        "Enable NSR.";
      leaf enable {
        type boolean;
      }
    }

    container nsf {
      choice nsf-type {
        case cisco {
          if-feature nsf-cisco;
          description
            "Enable Cisco Non Stop Forwarding.";
          leaf cisco-enable {
            type boolean;
          }
        }
        case ietf {
          description
            "Enable Cisco Non Stop Forwarding.";
          leaf ietf-enable {
            type boolean;
          }
          leaf ietf-helper-enable {
            type boolean;
          }
        }
      }
    }

    container protocol {
      description
        "Protocol specific configuration.";
      leaf shutdown {
        type boolean;
      }
    }

    container spf {
      if-feature spf;
      description
        "SPF configuration";
      leaf prefix-priority-policy {
        type string;
      }
    }
  }
}
```

```
container snmp {
  description
    "Adjust ospf snmp parameters";
  leaf context {
    type string;
  }
}

container redistribute {
  description
    "Redistribute information from another
    routing protocol.
    This grouping is intended to be augmented by
    vendors to implement vendor-specific protocol
    redistribution configuration options.";
  choice protocol {
    case bgp {
      leaf enable-bgp {
        type boolean;
      }
    }
    case ospf {
      leaf enable-ospf {
        type boolean;
      }
    }
    case isis {
      leaf enable-isis {
        type boolean;
      }
    }
    case connected {
      leaf enable-connected {
        type boolean;
      }
    }
    case eigrp {
      leaf enable-eigrp {
        type boolean;
      }
    }
    case mobile {
      leaf enable-mobile {
        type boolean;
      }
    }
    case static {
      leaf enable-static {
```

```
        type boolean;
    }
}
case rip {
    leaf enable-rip {
        type boolean;
    }
}
}
}

container timers {
    description
        "Adjust routing timers.
        This grouping is intended to be augmented
        by vendors to implement vendor-specific
        protocol timers configuration options";

    container lsa {
        leaf min-arrival {
            description
                "The minimum interval in milliseconds
                between accepting the same";
            units milliseconds;
            type uint32 {
                range "0..600000";
            }
        }
        leaf refresh {
            description
                "The minimum interval in seconds
                between refresh";
            units seconds;
            type uint32 {
                range "1800..2700";
            }
        }
    }
}

container throttle-lsa {
    leaf delay {
        description
            "Delay to generate first occurrence of
            LSA in milliseconds";
        units milliseconds;
        type uint32 {
            range "0..600000";
        }
    }
}
```

```
leaf min-delay {
  description
    "The Minimum delay between originating
    the same LSA in milliseconds";
  units milliseconds;
  type uint32 {
    range "0..600000";
  }
}
leaf max-delay {
  description
    "The Maximum delay between originating
    the same LSA in milliseconds";
  units milliseconds;
  type uint32 {
    range "0..600000";
  }
}
}

container throttle-spf {
  leaf delay {
    description
      "Delay between receiving a change to
      SPF calculation in milliseconds";
    units milliseconds;
    type uint32 {
      range "0..600000";
    }
  }
  leaf min-delay {
    description
      "Delay between first and second SPF
      calculation in milliseconds";
    units milliseconds;
    type uint32 {
      range "0..600000";
    }
  }
  leaf max-delay {
    description
      "Maximum wait time in milliseconds
      for SPF calculations";
    units milliseconds;
    type uint32 {
      range "0..600000";
    }
  }
}
```

```
    }
  }

  container mpls {
    description
      "OSPF MPLS configuraions.";
    leaf te-rid {
      description
        "Traffic Engineering stable
         IP address for system.";
      type if:interface-ref;
    }
    leaf ldp-sync-enable {
      description
        "Enable LDP IGP synchronization.";
      type boolean;
    }
    leaf ldp-autoconfig-enable {
      description
        "Enable LDP IGP interface
         auto-configuration.";
      type boolean;
    }
  }
}

container fast-reroute {
  leaf fast-reroute-enable {
    description
      "IP fast reroute.";
    type boolean;
  }
}

container ospf-areas {
  description
    "The top level container for the list
     of areas of the OSPF router.";

  uses area-inherit-config;
  uses interface-inherit-config;

  list ospf-area {
    key "area-id";
    leaf area-id {
      type uint32;
    }

    leaf area-notation {
```

```
description
  "Sets the default area notation.";
type enumeration {
  enum "decimal";
  enum "dot";
}
}
choice area-type {
  case nssa {
    description
      "Specify area as a NSSA area.";
    leaf redistribute {
      type boolean;
    }
    leaf nssa-summary {
      type boolean;
    }
    leaf nssa-ext-capability {
      if-feature nssa-ext-capability;
      type boolean;
    }
  }
  container default-information-originate {
    description
      "Originate Type 7 default into NSSA area.";
    leaf metric {
      type uint16 {
        range "1..65535";
      }
    }
    leaf metric-type {
      type uint8 {
        range "1..2";
      }
    }
  }
}
}
case stub {
  description
    "Specify area as a stub area.";
  leaf stub-summary {
    type boolean;
  }
  leaf stub-ext-capability {
    type boolean;
  }
}
}
```

```
leaf default-cost {
  description
    "Set the summary default-cost of a
    NSSA/stub area.";
  type uint32 {
    range "1..16777215";
  }
}

list range {
  description
    "Summarize routes matching
    address/mask (border routers only)";
  key "prefix";
  leaf prefix {
    type inet:ip-prefix;
  }
  leaf advertise {
    type boolean;
  }
  leaf cost {
    type uint32 {
      range "0..16777214";
    }
  }
}

container virtual-link {
  description
    "Define a virtual link";
  leaf router-id {
    type inet:ip-address;
  }

  uses interface-internal-inherit-config;
}

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

```
        uses interface-internal-inherit-config;
    }

    container mpls-te-config {
        leaf mpls-te-enable {
            description
                "Enable an ospf area to run MPLS
                Traffic Engineering.";
            type boolean;
        }
    }

    container ospf-interfaces {
        description
            "The top level container for the
            list of interfaces of the OSPF router.";

        uses interface-inherit-config;

        list ospf-interface {
            key "interface";
            leaf interface {
                type if:interface-ref;
            }
            uses interface-inherit-config;
        } // list ospf-interface
    } // container ospf-interfaces
} // list ospf-area
} // container ospf-areas
} // container ospf-af
} // container ospf-afs
} // list ospf-router
} // container ospf-routers
}
<CODE ENDS>
```

4. Security Considerations

The data model defined does not create any security implications.

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

5. Acknowledgements

The authors would like to thank Gaurav Gupta and many others for review and comments.

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

6. References

6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.
- [RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

6.2. Informative References

- [I-D.ietf-netmod-interfaces-cfg]
Bjorklund, M., "A YANG Data Model for Interface Management", draft-ietf-netmod-interfaces-cfg-12 (work in progress), July 2013.
- [I-D.ietf-netmod-routing-cfg]
Lhotka, L., "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-11 (work in progress), October 2013.

Authors' Addresses

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

E-Mail: myeung@cisco.com

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

E-Mail: yiqu@cisco.com

Alexander Clemm
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
USA

EMail: alex@cisco.com