LSVR
Internet-Draft
Intended status: Standards Track
Expires: 13 August 2026

M. Jethanandani, Ed.
K. Patel
Arrcus, Inc
A. MahendraBabu
Cisco, Inc

9 February 2026

A YANG Model for BGP-LS, BGP-LS-VPN, and BGP-LS-SPF
draft-ietf-lsvr-bgp-ls-yang-03

Abstract

This document defines a YANG data model for configuration and management of BGP-LS, BGP-LS-VPN, and BGP-LS-SPF.

Status of This Memo

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

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

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

This Internet-Draft will expire on 13 August 2026.

Table of Contents

1. Introduction

North-Bound Distribution of Link-State (LS) and Traffic Engineering (TE) Information Using BGP [RFC9552] describes a mechanism by which LS and TE information can be collected and shared with external components using BGP routing protocol. That LS combined with Shortest Path First (SPF) algorithm can be used by BGP for making routing decisions. Additionally, BGP Link-State Shortest Path First (SPF) Routing [RFC9815] describes how it allows BGP to be used efficiently as both the underlay and the overlay protocol in Many Massively Scaled Data Centers (MSDC). This document defines a YANG 1.1 [RFC7950] model that can be used to configure a router of that capability. It also defines a model for Link State DataBase (LSDB) that is used to store Link State Advertisements (LSA).

The model conforms to the NMDA [RFC8342] architecture.

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.

2. Terminology

This document references terms defined in other documents.

* BGP SPF Routing Domain

* BGP-LS-SPF NLRI

2.1. Acronyms

This document uses a few acronyms. Some of them are defined here for reference.

Acronym	Definition
AFI	Address Family Indicator
LS	Link-State
LSA	Link-State Advertisements
LSDB	Link-State DataBase
SAFI	Subsequent Address Family Indicator
SPF	Shortest Path First
TE	Traffic Engineering

Table 1: Acronyms

3. Tree Diagram

An abridged version of the tree diagram is shown here. Annotations used in the diagram are defined in YANG Tree Diagrams [RFC8340].

```
module: ietf-bgp-ls

  augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/bgp:bgp/bgp:global
            /bgp:afi-safis/bgp:afi-safi:
    +--rw link-state!
    |  +--rw prefix-limit
    |  |  +--rw max-prefixes?            uint32
    |  |  +--rw warning-threshold-pct?   rt-types:percentage
    |  |  +--rw teardown?                boolean
```

```
      |  |  +--rw idle-time?               union
      |  |  +--ro prefix-limit-exceeded?   boolean
      |  +--rw max-rate?          uint32
      |  +--rw max-number?        uint32
      |  +--ro bgp-ls-topology
      |     +--ro instances
      |           ...
      +--rw link-state-spf
         +--rw prefix-limit
         |  +--rw max-prefixes?            uint32
         |  +--rw warning-threshold-pct?   rt-types:percentage
         |  +--rw teardown?                boolean
         |  +--rw idle-time?               union
         |  +--ro prefix-limit-exceeded?   boolean
         +--rw instance-identifier?   uint64
         +--rw algorithm-type?        spf-algorithm-type
         +--rw node-status?           enumeration
         +--ro log
         |  +--ro event* (id)
         |        ...
         +--ro bgp-ls-topology
            +--ro instances
                  ...
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/bgp:bgp/bgp:neighbors
             /bgp:neighbor/bgp:afi-safis/bgp:afi-safi:
     +--rw link-state!
     |  +--rw prefix-limit
     |  |  +--rw max-prefixes?            uint32
     |  |  +--rw warning-threshold-pct?   rt-types:percentage
     |  |  +--rw teardown?                boolean
     |  |  +--rw idle-time?               union
     |  |  +--ro prefix-limit-exceeded?   boolean
     |  +--rw max-rate?       uint32
     |  +--rw max-number?     uint32
     +--rw link-state-spf
        +--rw prefix-limit
        |  +--rw max-prefixes?            uint32
        |  +--rw warning-threshold-pct?   rt-types:percentage
        |  +--rw teardown?                boolean
        |  +--rw idle-time?               union
        |  +--ro prefix-limit-exceeded?   boolean
        +--rw metric?         uint32
        +--rw status?         enumeration
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/bgp:bgp/bgp:neighbors
             /bgp:neighbor/bgp:statistics:
     +--ro updates-sent?             yang:zero-based-counter32
```

```
      +--ro updates-received?         yang:zero-based-counter32
      +--ro error-updates-received?   yang:zero-based-counter32
      +--ro computations?             yang:zero-based-counter32
      +--ro triggering-events?        yang:zero-based-counter32
      +---x clear {bt:clear-statistics}?
         +---w input
         |  +---w clear-at?   yang:date-and-time
         +--ro output
            +--ro clear-finished-at?   yang:date-and-time
    augment /rt:routing/rt:control-plane-protocols
              /rt:control-plane-protocol/bgp:bgp/bgp:peer-groups
              /bgp:peer-group/bgp:afi-safis/bgp:afi-safi:
      +--rw link-state!
      |  +--rw prefix-limit
      |  |  +--rw max-prefixes?            uint32
      |  |  +--rw warning-threshold-pct?   rt-types:percentage
      |  |  +--rw teardown?                boolean
      |  |  +--rw idle-time?               union
      |  |  +--ro prefix-limit-exceeded?   boolean
      |  +--rw max-rate?       uint32
      |  +--rw max-number?     uint32
      +--rw link-state-spf
         +--rw prefix-limit
         |  +--rw max-prefixes?            uint32
         |  +--rw warning-threshold-pct?   rt-types:percentage
         |  +--rw teardown?                boolean
         |  +--rw idle-time?               union
         |  +--ro prefix-limit-exceeded?   boolean
         +--rw metric?         uint32
         +--rw status?         enumeration
```

Figure 1: Tree Diagram for BGP-LS, BGP-LS-SPF YANG Model

4. YANG Models

4.1. BGP Link-State YANG model

The YANG model augments the BGP model in BGP Model for Service Provider Network [I-D.ietf-idr-bgp-model] to add extensions to BGP configuration. These extensions include the addition of three new Address Family Indicator (AFI) and Subsequent Address Family Indicator (SAFI) - BGP-LS, BGP-LS-VPN, and BGP-LS-SPF.

The BGP model is augmented both at a global level, and at a neighbor level to add support for the three new AFI/SAFI. In addition, there is support for statistics both at a global and at a neighbor level for the new address families. The 'feature' definition in BGP Model for Service Provider Network [I-D.ietf-idr-bgp-model] for 'clear-statistics' is used in this model to determine whether ability to clear statistics will be supported.

The model imports Common YANG Data Types [RFC9911], A YANG Data Model for Routing Management(NMDA Version) [RFC8349], and BGP Model for Service Provider Network [I-D.ietf-idr-bgp-model].

<CODE BEGINS> file "ietf-bgp-ls@2026-02-09.yang"

```
module ietf-bgp-ls {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-bgp-ls";
  prefix bgp-ls;

  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 9911: Common YANG Data Types.";
  }
  import ietf-routing {
    prefix rt;
    reference
      "RFC 8349, A YANG Data Model for Routing Management
       (NMDA Version).";
  }
  import ietf-bgp {
    prefix bgp;
    reference
      "I-D.ietf-idr-bgp-model: BGP YANG Model for Service Provider
                               Networks.";
  }
  import iana-bgp-types {
    prefix bt;
    reference
      "I-D.ietf-idr-bgp-model: BGP YANG Model for Service Provider
                               Networks.";
  }
  import ietf-bgp-lsdb {
    prefix bgp-lsdb;
    reference
      "RFC XXXX: A YANG model for BGP-LS, BGP-LS-VPN,
                 and BGP-LS-SPF.";
  }
```

```
     organization
       "IETF LSVR Working Group";
     contact
       "WG Web:   <https://datatracker.ietf.org/wg/lsvr>
        WG List:  <lsvr@ietf.org>

        Authors: Mahesh Jethanandani (mjethanandani at gmail.com),
                 Keyur Patel (keyur at arrcus.com),
                 Aravind Babu MahendraBabu (aramahen at cisco.com)";

     description
       "This module contains contains management
        information for BGP-LS database.

        Copyright (c) 2023 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 XXXX
        (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
        for full legal notices.

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

     revision 2026-02-09 {
       description
         "Initial Version";
       reference
         "RFC XXXX, BGP Model for Link State Distribution.";
     }

     // Identities.
     identity bgp-ls {
       base bt:afi-safi-type;
       description
         "BGP Link-State.";
       reference
         "RFC 9552: Distribution of Link-State and Traffic Engineering
```

```
          Information Using BGP.";
      }

      identity bgp-ls-vpn {
        base bt:afi-safi-type;
        description
          "BGP Link-State VPN.";
        reference
          "RFC 9552: Distribution of Link-State and Traffic Engineering
           Information Using BGP.";
      }

      identity bgp-ls-spf {
        base bt:afi-safi-type;
        description
          "BGP Link-State Shortest Path First (BGP-LS-SPF).";
        reference
          "RFC 9815: BGP Link-State SPF Routing.";
      }

      identity attribute-type {
        description
          "Base identity for BGP-LS Attribute type.";
      }

      identity multi-topology-identifier {
        base attribute-type;
        description
          "Multi Topology Identifier TLV.";
        reference
          "RFC 9552: Distribution of Link-State and Traffic Engineering
           Information Using BGP.";
      }

      identity node-flag-bits {
        base attribute-type;
        description
          "The Node Flag Bits TLV.";
        reference
            "RFC 9552: Distribution of Link-State and Traffic
             Engineering Information Using BGP.";
      }

      identity opaque-node-attribute {
        base attribute-type;
        description
          "The Opaque Node Attribute TLV.";
        reference
```

```
        "RFC 9552: Distribution of Link-State and Traffic Engineering
         Information Using BGP.";
    }

    identity node-name {
      base attribute-type;
      description
        "The Node Name TLV.";
      reference
        "RFC 9552: Distribution of Link-State and Traffic Engineering
         Information Using BGP.";
    }

    identity isis-area-identifier {
      base attribute-type;
      description
        "IS-IS Area Identifier.";
      reference
        "RFC 9552: Distribution of Link-State and Traffic Engineering
         Information Using BGP.";
    }

    identity ipv4-route-id-of-local-node {
      base attribute-type;
      description
        "IPv4 Router-ID of Local Node.";
      reference
        "RFC 9552: Distribution of Link-State and Traffic Engineering
         Information Using BGP.";
    }

    identity ipv6-router-id-of-local-node {
      base attribute-type;
      description
        "IPv6 Router-ID of Local Node.";
      reference
        "RFC 9552: Distribution of Link-State and Traffic Engineering
         Information Using BGP.";
    }

    // Typedefs
    typedef spf-algorithm-type {
      type enumeration {
        enum normal {
          description
            "Normal Shortest Path First (SPF) algorithm based on link
             metric. This is the standard shortest path algorithm as
             computed by the IGP protocol. Consistent with the
```

```
            deployed practice for link-state protocols, Algorithm 0
            permits any node to overwrite the SPF path with a
            different path based on its local policy.";
        }
        enum strict {
          description
            "Strict Shortest Path First (SPF) algorithm based on link
             metric. The algorithm is identical to Algorithm 0 but
             Algorithm 1 requires that all nodes along the path will
             honor the SPF routing decision. Local policy at the node
             claiming support for Algorithm 1 MUST NOT alter the SPF
             paths computed by Algorithm 1";
        }
        enum unknown {
          description
            "Unknown Algorithm";
        }
      }
      description
        "SPF algorithm type.";
    }

    // Groupings
    grouping bgp-neighbor-ls-common {
      description
        "Grouping for neighbor configuration for Link-State.";

        leaf metric {
          type uint32;
          default 10;
          description
            "Metric associated with the corresponding link to
             be used in the SPF graph computation.";
        }

        leaf status {
          type enumeration {
            enum reachable {
              description
                "The link is reachable in the current SPF topology.";
            }
            enum uncreachable {
              description
                "The link is unreachable in the current SPF
                 topology.";
            }
          }
          default reachable;
```

```
          description
            "Sets SPF-Status of the corresponding LS Link NLRI.";

        }
      }

      grouping bgp-mp-ls {
        description
          "Grouping for BGP-LS paramters.";

        container link-state {
          when "derived-from-or-self(../../bgp:afi-safi/bgp:name, " +
               "'bgp-ls')" {
                  description
                    "Include this container for BGP Linkstate specific
               configuration";
          }
          presence
                  "This container is for BGP Linkstate specific
                  congfiguration.";

          description
            "Information related to Link-State configuration and
             management.";

          uses bgp:mp-all-afi-safi-common;

          leaf max-rate {
            type uint32;
            units per-second;
            default 200;
            description
              "Maximum rate at which Link-State NLRIs will be
               advertised or withdrawn from neighbors.";
            reference
              "RFC 9552: Distribution of Link-State and Traffic
               Engineering Information Using BGP.";
          }

          leaf max-number {
            type uint32;
            description
              "Maximim number of Link-State NLRIs stored in a router's
               RIB.";
            reference
              "RFC 9552: Distribution of Link-State and Traffic
               Engineering Information Using BGP.";
          }
```

```
      }
    }

    grouping bgp-mp-ls-spf {
      description
        "Grouping for BGP-LS-SPF parameters.";

      container link-state-spf {
        when "derived-from-or-self (../../bgp:afi-safi/bgp:name, " +
                                   "'bgp-ls-spf')" {
          description
            "Include this container for BGP Link-State SPF specific
             configuration";
        }
        description "BGP Linkstate-SPF configuration options";

        uses bgp:mp-all-afi-safi-common;
      }
    }

    augment "/rt:routing/rt:control-plane-protocols" +
            "/rt:control-plane-protocol/bgp:bgp/bgp:global" +
            "/bgp:afi-safis/bgp:afi-safi" {
      description
        "Augmentation of the BGP model to add BGL-LS.";
      uses bgp-mp-ls;
      uses bgp-mp-ls-spf;
    }

    augment "/rt:routing/rt:control-plane-protocols" +
            "/rt:control-plane-protocol/bgp:bgp/bgp:global" +
            "/bgp:afi-safis/bgp:afi-safi/link-state" {
      description
        "Augmentation of BGL-LS to add dB.";
      uses bgp-lsdb:lsdb-top;
    }

    augment "/rt:routing/rt:control-plane-protocols" +
            "/rt:control-plane-protocol/bgp:bgp/bgp:global" +
            "/bgp:afi-safis/bgp:afi-safi/link-state-spf" {
      description
        "Augmentation of BGL-LS-SPF.";

      leaf instance-identifier {
        type uint64;
        default 0;
        description
          "Instance Identifier to be used for all Link-State NLRI
```

```
          advertisements originated locally";
      }

      leaf algorithm-type {
        type spf-algorithm-type;
        default normal;
        description
          "SPF Algorithm type associated with Link-State AFI SAFI";
      }

      leaf node-status {
        type enumeration {
          enum reachable {
            description
              "The local node is reachable in the current SPF
               topology.";
          }
          enum unreacable {
            description
              "The local node is unreachable in the current SPF
               topology.";
          }
          enum no-transit-support {
            description
              "The local node is reachable but does not support
               forwarding of transit traffic.";
          }
        }
        default reachable;
        description
          "Sets SPF-Status of the local node.";
      }

      container log {
        config false;
        description
          "This container lists the SPF computation events.";

        list event {
          key id;
          description
            "List of computation events - implemented as a
             wrapping buffer.";

          leaf id {
            type uint32;
            description
              "Event identifier.";
```

```
            }

            leaf type {
              type enumeration {
                enum full {
                  description "Full SPF computation.";
                }
                enum route-only {
                  description
                    "Route reachability only SPF computation";
                }
              }
              description "Type of SPF computation performed.";
            }

            leaf schedule-time {
              type yang:date-and-time;
              description
                "Time when the SPF computation was
                 scheduled.";
            }

            leaf delay {
              type uint64;
              description
                "Delay in micro-seconds applied for this SPF event.";
            }

            leaf start-time {
              type yang:date-and-time;
              description
                "Time when the SPF computation started.";
            }

            leaf end-time {
              type yang:date-and-time;
              description
                "Time when the SPF computation ended.";
            }

            leaf duration {
              type uint64;
              description
                "Time taken in micro-seconds to execute the SPF
                 computations.";
            }

            leaf node-count {
```

```
            type uint64;
            description
              "Number of nodes involved in the SPF computations.";
          }

          leaf prefix-count {
            type uint64;
            description
              "Number of prefixes involved in the SPF computations.";
          }

          leaf route-download-count {
            type uint64;
            description
              "Number of routes updated in the SPF computations.";
          }

          list lsp-trigger {
            key "id";
            description
              "This list includes the LSPs that triggered the
               SPF computation.";

            leaf id {
              type uint32;
              description
                "Trigger identifier.";
            }

            leaf nlri-prefix {
              type string;
              description
                "Prefix of the NLRI triggering SPF computation.";
            }

            leaf nlri-sequence {
              type uint32;
              description
                "Sequence number of the NLRI triggering SPF
                 computation.";
            }

            leaf trigger-time {
              type yang:date-and-time;
              description
                "Time when the trigger event was recorded.";
            }
          }
```

```
        }
      }
      uses bgp-lsdb:lsdb-top;
    }

    augment "/rt:routing/rt:control-plane-protocols" +
            "/rt:control-plane-protocol/bgp:bgp/bgp:neighbors" +
            "/bgp:neighbor/bgp:afi-safis/bgp:afi-safi" {
      description
        "Augmentation of the BGP model to add BGL-LS.";
      uses bgp-mp-ls;
      uses bgp-mp-ls-spf;
    }

    augment "/rt:routing/rt:control-plane-protocols" +
            "/rt:control-plane-protocol/bgp:bgp/bgp:neighbors" +
            "/bgp:neighbor/bgp:afi-safis/bgp:afi-safi/link-state-spf" {
      description
        "Augmentation of the BGP neighbor to add BGL-LS.";
      uses bgp-neighbor-ls-common;
    }

    /*
    augment "/rt:routing/rt:control-plane-protocols" +
            "/rt:control-plane-protocol/bgp:bgp/bgp:global" +
            "/bgp:afi-safis/bgp:afi-safi/bgp:statistics" {
      when "derived-from-or-self (../bgp:name, 'bgp-ls')" {
        description
          "Include these for BGP Link-State specific statistics.";
      }
      description
        "Augmentation of the global statistics counter to add
         BGP-LS statistics.";

      leaf updates-sent {
        type yang:zero-based-counter32;
        description
          "Total number of Link-State NLRI updates sent.";
        reference
          "RFC 9552: Distribution of Link-State and Traffic
           Engineering Information Using BGP.";
      }

      leaf updates-received {
        type yang:zero-based-counter32;
        description
          "Total number of Link-State NLRI updates received.";
        reference
```

```
          "RFC 9552: Distribution of Link-State and Traffic
           Engineering Information Using BGP.";
      }

      leaf local-ls-originated {
        type yang:zero-based-counter32;
        description
          "Total number of locally originated Link-State NLRIs.";
        reference
          "RFC 9552: Distribution of Link-State and Traffic
           Engineering Information Using BGP.";
      }

      action clear {
        if-feature "bt:clear-statistics";
        description
          "Clear statistics action command.

          Execution of this command should result in all the
          counters to be cleared and set to 0.";

        input {
          leaf clear-at {
            type yang:date-and-time;
            description
              "Time when the clear action needs to be executed.";
          }
        }

        output {
          leaf clear-finished-at {
            type yang:date-and-time;
              description
                "Time when the clear action command completed.";
          }
        }
      }
    }
    augment "/rt:routing/rt:control-plane-protocols" +
            "/rt:control-plane-protocol/bgp:bgp/bgp:global" +
            "/bgp:afi-safis/bgp:afi-safi/bgp:statistics" {
      when "derived-from-or-self (../bgp:name, 'bgp-ls-spf')" {
        description
          "Include these for BGP Link-State SPF specific statistics.";
      }
      description
        "Augmentation of the global statistics counter to add
         BGP-LS-SPF statistics.";
```

```
        leaf scheduled {
          type yang:zero-based-counter64;
          description
            "Number of times SPF has been re/scheduled";
        }

        leaf computed {
          type yang:zero-based-counter64;
          description
            "Number of times SPF has been computed";
        }

        leaf maximum-duration {
          type uint64;
          units "microseconds";
          description
            "Maximum duration taken for SPF compuation in
             microseconds.";
        }

        leaf minimum-duration {
          type uint64;
          units "microseconds";
          description
            "Minimum duration taken for SPF compuation in
             microseconds.";
        }

        leaf average-duration {
          type uint64;
          units "microseconds";
          description
            "Average duration taken for SPF compuation in
             microseconds.";
        }

        leaf last-computation-time {
          type yang:date-and-time;
          description
            "Timestamp for last SPF computation for this
             address-family";
        }
      }
      */
      augment "/rt:routing/rt:control-plane-protocols" +
              "/rt:control-plane-protocol/bgp:bgp/bgp:neighbors" +
              "/bgp:neighbor/bgp:statistics" {
        when "derived-from-or-self (../bgp:afi-safis/bgp:afi-safi" +
```

```
                                "/bgp:name, 'bgp-ls')" {
        description
          "Include these for BGP Link-State specific statistics.";
      }
      description
        "Augmentation of the BGP per-neighbor statistics to add
         BGP-LS specific counters.";

      leaf updates-sent {
        type yang:zero-based-counter32;
        description
          "Total number of Link-State NLRIs updates sent per
           neighbor.";
        reference
          "RFC 9552: Distribution of Link-State and Traffic
           Engineering Information Using BGP.";
      }

      leaf updates-received {
        type yang:zero-based-counter32;
        description
          "Total number of Link-State NLRIs updates received per
           neighbor.";
        reference
          "RFC 9552: Distribution of Link-State and Traffic
           Engineering Information Using BGP.";
      }

      leaf error-updates-received {
        type yang:zero-based-counter32;
        description
          "Total number of Link-State NLRIs updates received that
           were errored, per neighbor.";
        reference
          "RFC 9552: Distribution of Link-State and Traffic
           Engineering Information Using BGP.";
      }

      leaf computations {
        type yang:zero-based-counter32;
        description
          "Count of number of SPF computations made.";
          reference
            "RFC 9815: BGP Link-State SPF Routing.";
      }

      leaf triggering-events {
        type yang:zero-based-counter32;
```

```
        description
          "SPF triggering events.";
          reference
            "RFC 9815: BGP Link-State SPF Routing.";
      }

      action clear {
        if-feature "bt:clear-statistics";
        description
          "Clear statistics action command.

          Execution of this command should result in all the
          counters to be cleared and set to 0.";

        input {
          leaf clear-at {
            type yang:date-and-time;
            description
              "Time when the clear action needs to be executed.";
          }
        }

        output {
          leaf clear-finished-at {
            type yang:date-and-time;
              description
                "Time when the clear action command completed.";
          }
        }
      }
    }

    augment "/rt:routing/rt:control-plane-protocols" +
            "/rt:control-plane-protocol/bgp:bgp/bgp:peer-groups" +
            "/bgp:peer-group/bgp:afi-safis/bgp:afi-safi" {
      description
        "Augmentation of the BGP peer-group to add BGP-LS and
         BGP-LS-SPF.";

      uses bgp-mp-ls;
      uses bgp-mp-ls-spf;
    }

    augment "/rt:routing/rt:control-plane-protocols" +
            "/rt:control-plane-protocol/bgp:bgp/bgp:peer-groups" +
            "/bgp:peer-group/bgp:afi-safis/bgp:afi-safi/" +
            "link-state-spf" {
      description
```

```
        "Augmentation of the BGP peer-groups to add BGL-LS.";
      uses bgp-neighbor-ls-common;
    }
  }
  <CODE ENDS>
```

Figure 2: YANG Model for BGP-LS, BGP-LS-VPN, BGP-LS-SPF

4.2. BGP Link-State Database YANG model

The model imports Common YANG Data Types [RFC9911]

```
<CODE BEGINS> file "ietf-bgp-lsdb@2026-02-09.yang"
module ietf-bgp-lsdb {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-bgp-lsdb";
  prefix bgp-lsdb;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 9911: Common YANG Data Types.";
  }

  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 9911: Common YANG Data Types.";
  }

  import ietf-bgp-ls-topo-types {
    prefix bgp-ls-topo-t;
    reference
      "RFC XXXX: BGP Model for Link State Distribution.";
  }

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

  import ietf-isis {
    prefix ietf-isis-t;
    reference
      "RFC 9130: YANG Data Model for IS-IS Protocol.";
  }
```

```
  import ietf-ospf {
    prefix ietf-ospf-t;
    reference
      "RFC 9129: YANG Data Model for the OSPF Protocol.";
  }

  include ietf-bgp-ls-topo-attr;

  organization
    "IETF LSVR Working Group";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/lsvr>
     WG List:  <lsvr@ietf.org>

     Authors: Mahesh Jethanandani (mjethanandani at gmail.com),
              Keyur Patel (keyur at arrcus.com),
              Aravind Babu MahendraBabu (aramahen at cisco.com)";

  description
    "This module contains contains configuration and management
     information for BGP-LS, BGP-LS-SPF.

     Copyright (c) 2023 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 XXXX
     (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
     for full legal notices.

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

  revision 2026-02-09 {
    description
      "Initial Version";
    reference
      "RFC XXXX, BGP Model for Link State Distribution.";
  }
```

```
     // Groupings

     grouping lsdb-top {
       description
         "Details for all databases under an application instance.";

       container bgp-ls-topology {
         config false;
         description
           "Top level container for BGP-LS topology entries";

         container instances {
           description
             "Enclosing container for instance list";

           list instance {
             key "vrf-name protocol identifier";
             description
               "List of instances";

             leaf vrf-name {
               type string;
               description
                 "Name of the VRF this database belongs to.";
             }

             leaf protocol {
               type bgp-ls-topo-t:protocol;
               description
                 "Protocol originating the NLRI.";
               reference
                 "RFC 9552: Distribution of Link-State and Traffic
                  Engineering Information Using BGP, Section 5.2.";
             }

             leaf identifier {
               type uint64;
               description
                 "Instance identifier identifying the IGP routing
                  domain where the NLRI belongs.";
               reference
                 "RFC 9552: Distribution of Link-State and Traffic
                  Engineering Information Using BGP, Section 5.2.";
             }

             container nodes {
               description
                 "BGP-LS topology entries for nodes";
```

```
            list ospf-node {
              key "is-as-scoped area-id router-id dr-identifier as";
              description
                "List of node types";

              leaf is-as-scoped {
                type boolean;
                description
                  "Is area ID valid?";
              }

              leaf area-id {
                type ietf-ospf-t:area-id-type;
                description
                  "Area ID.";
                reference
                  "RFC 9552: Distribution of Link-State and
                   Traffic Engineering Information Using BGP,
                   Section 5.2.1.";
              }

              leaf router-id {
                type ietf-rt-t:router-id;
                description
                  "Router ID.";
                reference
                  "RFC 9552: Distribution of Link-State and
                   Traffic Engineering Information Using BGP,
                   Section 5.2.1.";
              }

              leaf dr-identifier {
                type inet:ipv4-address;
                description
                  "Designated Router (DR) Identifier.";
                reference
                  "RFC 9552: Distribution of Link-State and
                   Traffic Engineering Information Using BGP,
                   Section 5.2.1.";
              }

              leaf as {
                type inet:as-number;
                description
                  "Autonomous System (AS) number associated with the
                   BGP process originating the NLRI.";
                reference
                  "RFC 9552: Distribution of Link-State and
```

```
                    Traffic Engineering Information Using BGP,
                    Section 5.2.1.";
                }

                uses bgp-ls-topo-node-attr;
              }

              list isis-node {
                key "system-id psn-id as";
                description
                  "List of node types";

                leaf system-id {
                  type ietf-isis-t:system-id;
                  description
                    "System ID.";
                  reference
                    "RFC 9552: Distribution of Link-State and
                     Traffic Engineering Information Using BGP,
                     Section 5.2.1.";
                }

                leaf psn-id {
                  type uint8;
                  description
                    "Pseudo Node Identifier (PSN ID).";
                  reference
                    "RFC 9552: Distribution of Link-State and
                     Traffic Engineering Information Using BGP,
                     Section 5.2.1.";
                }

                leaf as {
                  type inet:as-number;
                  description
                    "Autonomous System (AS) number associated with the
                     BGP process originating the NLRI.";
                  reference
                    "RFC 9552: Distribution of Link-State and
                     Traffic Engineering Information Using BGP,
                     Section 5.2.1.";
                }

                uses bgp-ls-topo-node-attr;
              }
            }

            container unknowns {
```

```
                description
                  "BGP-LS topology entries for unknown NLRIs -- active
                  when the nlri-type is unknown";
                list unknown {
                  key "nlri";
                  description
                    "List of unknown types";

                  leaf nlri {
                    type yang:hex-string;
                    description
                      "NLRI.";
                    reference
                      "RFC 9552: Distribution of Link-State and Traffic
                       Engineering Information Using BGP, Section 5.2.";
                  }

                  leaf attributes {
                    type yang:hex-string;
                    description
                      "Attributes.";
                    reference
                      "RFC 9552: Distribution of Link-State and Traffic
                       Engineering Information Using BGP, Section 5.3.";
                  }
                }
              }
            }
          }
        }
      }
    }
  }
  <CODE ENDS>
```

Figure 3: YANG Model for Link-State Database

5. IANA Considerations

This memo registers the following namespace URIs in the IETF XML in the "IETF XML Registry" [RFC3688]:

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

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

This document registers the following YANG modules in the "YANG Module Names" registry [RFC6020]:

Name: ietf-bgp-ls
Namespace: urn:ietf:params:xml:ns:yang:ietf-bgp-ls
Prefix: bgp-ls
Reference: RFC XXXX

Name: ietf-bgp-lsdb
Namespace: urn:ietf:params:xml:ns:yang:ietf-bgp-lsdb
Prefix: bgp-lsdb
Reference: RFC XXXX

6. 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:

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:

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:

7. References

7.1. Normative References

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

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

[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>.

[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>.

[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>.

[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>.

[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>.

[RFC9552] Talaulikar, K., Ed., "Distribution of Link-State and Traffic Engineering Information Using BGP", RFC 9552, DOI 10.17487/RFC9552, December 2023, <https://www.rfc-editor.org/info/rfc9552>.

[RFC9815] Patel, K., Lindem, A., Zandi, S., and W. Henderickx, "BGP Link State (BGP-LS) Shortest Path First (SPF) Routing", RFC 9815, DOI 10.17487/RFC9815, July 2025, <https://www.rfc-editor.org/info/rfc9815>.

[RFC9911] Sch旦nw辰lder, J., Ed., "Common YANG Data Types", RFC 9911, DOI 10.17487/RFC9911, December 2025, <https://www.rfc-editor.org/info/rfc9911>.

[I-D.ietf-idr-bgp-model] Jethanandani, M., Patel, K., Hares, S., and J. Haas, "YANG Model for Border Gateway Protocol (BGP-4)", Work in Progress, Internet-Draft, draft-ietf-idr-bgp-model-18, 21 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-bgp-model-18>.

7.2. Informative References

Appendix A. Complete Tree Diagram

Here is a complete tree diagram for the configuration and operational part of the model.

```
module: ietf-bgp-ls

  augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/bgp:bgp/bgp:global
            /bgp:afi-safis/bgp:afi-safi:
    +--rw link-state!
```

```
     |  +--rw prefix-limit
     |  |  +--rw max-prefixes?            uint32
     |  |  +--rw warning-threshold-pct?   rt-types:percentage
     |  |  +--rw teardown?                boolean
     |  |  +--rw idle-time?               union
     |  |  +--ro prefix-limit-exceeded?   boolean
     |  +--rw max-rate?          uint32
     |  +--rw max-number?        uint32
     |  +--ro bgp-ls-topology
     |     +--ro instances
     |        +--ro instance* (vrf-name protocol identifier)
     |           +--ro vrf-name      string
     |           +--ro protocol      bgp-ls-topo-t:protocol
     |           +--ro identifier    uint64
     |           +--ro nodes
     |           |  +--ro ospf-node*
     |           |  |       (is-as-scoped area-id router-id dr-identifie
 r as)
     |           |  |  +--ro is-as-scoped       boolean
     |           |  |  +--ro area-id
     |           |  |  |       ietf-ospf-t:area-id-type
     |           |  |  +--ro router-id          ietf-rt-t:router-id
     |           |  |  +--ro dr-identifier      inet:ipv4-address
     |           |  |  +--ro as                 inet:as-number
     |           |  |  +--ro node-attributes
     |           |  |     +--ro node-attribute* (type)
     |           |  |        +--ro type
     |           |  |        |       identityref
     |           |  |        +--ro local-ipv4-router-ids
     |           |  |        |  +--ro router-id*   ietf-rt-t:router-id
     |           |  |        +--ro local-ipv6-router-ids
     |           |  |        |  +--ro router-id*   inet:ipv6-address
     |           |  |        +--ro unknowns
     |           |  |        |  +--ro unknown* (type)
     |           |  |        |     +--ro type     uint16
     |           |  |        |     +--ro value?   yang:hex-string
     |           |  |        +--ro multi-topology-ids
     |           |  |        |  +--ro multi-topology-id*   uint16
     |           |  |        +--ro node-flag-bits
     |           |  |        |  +--ro flags?
     |           |  |        |          bgp-ls-topo-t:node-flag-bits
     |           |  |        +--ro opaque-node-attribute
     |           |  |        |  +--ro opaque-attrs?   yang:hex-string
     |           |  |        +--ro node-name
     |           |  |        |  +--ro name?   string
     |           |  |        +--ro sr-capabilities
     |           |  |        |  +--ro srgb* (start-label)
     |           |  |        |  |  +--ro start-label
```

```
     |           |  |        |  |  |       ietf-rt-t:mpls-label
     |           |  |        |  |  +--ro range-size?    uint32
     |           |  |        |  +--ro isis-flags?   binary
     |           |  |        +--ro sr-local-block
     |           |  |        |  +--ro srlb* (start-label)
     |           |  |        |  |  +--ro start-label
     |           |  |        |  |  |       ietf-rt-t:mpls-label
     |           |  |        |  |  +--ro range-size?    uint32
     |           |  |        |  +--ro isis-flags?   binary
     |           |  |        +--ro sr-algorithm
     |           |  |        |  +--ro algorithm*   uint8
     |           |  |        +--ro node-msd
     |           |  |        |  +--ro msd* (msd-type)
     |           |  |        |     +--ro msd-type     uint8
     |           |  |        |     +--ro msd-value?   uint8
     |           |  |        +--ro fads
     |           |  |        |  +--ro fad* (algorithm)
     |           |  |        |     +--ro algorithm              uint8
     |           |  |        |     +--ro metric-type?           uint8
     |           |  |        |     +--ro calc-type?             uint8
     |           |  |        |     +--ro priority?              uint8
     |           |  |        |     +--ro node-fad-attributes
     |           |  |        |        +--ro node-fad-attribute* (type)
     |           |  |        |           +--ro type
     |           |  |        |           |       identityref
     |           |  |        |           +--ro exclude-any-affinity
     |           |  |        |           |  +--ro eag*   uint32
     |           |  |        |           +--ro include-any-affinity
     |           |  |        |           |  +--ro eag*   uint32
     |           |  |        |           +--ro include-all-affinity
     |           |  |        |           |  +--ro eag*   uint32
     |           |  |        |           +--ro flags
     |           |  |        |           |  +--ro flags?   binary
     |           |  |        |           +--ro exclude-srlg
     |           |  |        |           |  +--ro srlg*   uint32
     |           |  |        |           +--ro exclude-minimum-bw
     |           |  |        |           |  +--ro bw?
     |           |  |        |           |          ietf-rt-t:bandwidth-
ieee-float32
     |           |  |        |           +--ro exclude-maximum-delay
     |           |  |        |           |  +--ro delay?   uint32
     |           |  |        |           +--ro exclude-any-reverse-affin
ity
     |           |  |        |           |  +--ro eag*   uint32
     |           |  |        |           +--ro include-any-reverse-affin
ity
     |           |  |        |           |  +--ro eag*   uint32
     |           |  |        |           +--ro include-all-reverse-affin
```

```
 ity
     |           |  |        |           |  +--ro eag*   uint32
     |           |  |        |           +--ro fa-unsupported-tlvs
     |           |  |        |              +--ro protocol-id?
     |           |  |        |              |       bgp-ls-topo-t:protoc
 ol
     |           |  |        |              +--ro type*
     |           |  |        |                      uint16
     |           |  |        +--ro isis-area-identifiers
     |           |  |           +--ro area-address*
     |           |  |                   ietf-isis-t:area-address
     |           |  +--ro isis-node* (system-id psn-id as)
     |           |     +--ro system-id          ietf-isis-t:system-id
     |           |     +--ro psn-id             uint8
     |           |     +--ro as                 inet:as-number
     |           |     +--ro node-attributes
     |           |        +--ro node-attribute* (type)
     |           |           +--ro type
     |           |           |       identityref
     |           |           +--ro local-ipv4-router-ids
     |           |           |  +--ro router-id*   ietf-rt-t:router-id
     |           |           +--ro local-ipv6-router-ids
     |           |           |  +--ro router-id*   inet:ipv6-address
     |           |           +--ro unknowns
     |           |           |  +--ro unknown* (type)
     |           |           |     +--ro type     uint16
     |           |           |     +--ro value?   yang:hex-string
     |           |           +--ro multi-topology-ids
     |           |           |  +--ro multi-topology-id*   uint16
     |           |           +--ro node-flag-bits
     |           |           |  +--ro flags?
     |           |           |          bgp-ls-topo-t:node-flag-bits
     |           |           +--ro opaque-node-attribute
     |           |           |  +--ro opaque-attrs?   yang:hex-string
     |           |           +--ro node-name
     |           |           |  +--ro name?   string
     |           |           +--ro sr-capabilities
     |           |           |  +--ro srgb* (start-label)
     |           |           |  |  +--ro start-label
     |           |           |  |  |       ietf-rt-t:mpls-label
     |           |           |  |  +--ro range-size?    uint32
     |           |           |  +--ro isis-flags?   binary
     |           |           +--ro sr-local-block
     |           |           |  +--ro srlb* (start-label)
     |           |           |  |  +--ro start-label
     |           |           |  |  |       ietf-rt-t:mpls-label
     |           |           |  |  +--ro range-size?    uint32
     |           |           |  +--ro isis-flags?   binary
```

```
     |           |           +--ro sr-algorithm
     |           |           |  +--ro algorithm*   uint8
     |           |           +--ro node-msd
     |           |           |  +--ro msd* (msd-type)
     |           |           |     +--ro msd-type     uint8
     |           |           |     +--ro msd-value?   uint8
     |           |           +--ro fads
     |           |           |  +--ro fad* (algorithm)
     |           |           |     +--ro algorithm              uint8
     |           |           |     +--ro metric-type?           uint8
     |           |           |     +--ro calc-type?             uint8
     |           |           |     +--ro priority?              uint8
     |           |           |     +--ro node-fad-attributes
     |           |           |        +--ro node-fad-attribute* (type)
     |           |           |           +--ro type
     |           |           |           |       identityref
     |           |           |           +--ro exclude-any-affinity
     |           |           |           |  +--ro eag*   uint32
     |           |           |           +--ro include-any-affinity
     |           |           |           |  +--ro eag*   uint32
     |           |           |           +--ro include-all-affinity
     |           |           |           |  +--ro eag*   uint32
     |           |           |           +--ro flags
     |           |           |           |  +--ro flags?   binary
     |           |           |           +--ro exclude-srlg
     |           |           |           |  +--ro srlg*   uint32
     |           |           |           +--ro exclude-minimum-bw
     |           |           |           |  +--ro bw?
     |           |           |           |          ietf-rt-t:bandwidth-
ieee-float32
     |           |           |           +--ro exclude-maximum-delay
     |           |           |           |  +--ro delay?   uint32
     |           |           |           +--ro exclude-any-reverse-affin
ity
     |           |           |           |  +--ro eag*   uint32
     |           |           |           +--ro include-any-reverse-affin
ity
     |           |           |           |  +--ro eag*   uint32
     |           |           |           +--ro include-all-reverse-affin
ity
     |           |           |           |  +--ro eag*   uint32
     |           |           |           +--ro fa-unsupported-tlvs
     |           |           |              +--ro protocol-id?
     |           |           |              |       bgp-ls-topo-t:protoc
ol
     |           |           |              +--ro type*
     |           |           |                      uint16
     |           |           +--ro isis-area-identifiers
```

```
     |           |              +--ro area-address*
     |           |                      ietf-isis-t:area-address
     |           +--ro unknowns
     |              +--ro unknown* (nlri)
     |                 +--ro nlri          yang:hex-string
     |                 +--ro attributes?   yang:hex-string
     +--rw link-state-spf
        +--rw prefix-limit
        |  +--rw max-prefixes?            uint32
        |  +--rw warning-threshold-pct?   rt-types:percentage
        |  +--rw teardown?                boolean
        |  +--rw idle-time?               union
        |  +--ro prefix-limit-exceeded?   boolean
        +--rw instance-identifier?   uint64
        +--rw algorithm-type?        spf-algorithm-type
        +--rw node-status?           enumeration
        +--ro log
        |  +--ro event* (id)
        |     +--ro id                      uint32
        |     +--ro type?                   enumeration
        |     +--ro schedule-time?          yang:date-and-time
        |     +--ro delay?                  uint64
        |     +--ro start-time?             yang:date-and-time
        |     +--ro end-time?               yang:date-and-time
        |     +--ro duration?               uint64
        |     +--ro node-count?             uint64
        |     +--ro prefix-count?           uint64
        |     +--ro route-download-count?   uint64
        |     +--ro lsp-trigger* (id)
        |        +--ro id                uint32
        |        +--ro nlri-prefix?      string
        |        +--ro nlri-sequence?    uint32
        |        +--ro trigger-time?     yang:date-and-time
        +--ro bgp-ls-topology
           +--ro instances
              +--ro instance* (vrf-name protocol identifier)
                 +--ro vrf-name      string
                 +--ro protocol      bgp-ls-topo-t:protocol
                 +--ro identifier    uint64
                 +--ro nodes
                 |  +--ro ospf-node*
                 |  |       (is-as-scoped area-id router-id dr-identifie
 r as)
                 |  |  +--ro is-as-scoped       boolean
                 |  |  +--ro area-id
                 |  |  |       ietf-ospf-t:area-id-type
                 |  |  +--ro router-id          ietf-rt-t:router-id
                 |  |  +--ro dr-identifier      inet:ipv4-address
```

```
|  |  +--ro as                 inet:as-number
|  |  +--ro node-attributes
|  |     +--ro node-attribute* (type)
|  |        +--ro type
|  |        |       identityref
|  |        +--ro local-ipv4-router-ids
|  |        |  +--ro router-id*   ietf-rt-t:router-id
|  |        +--ro local-ipv6-router-ids
|  |        |  +--ro router-id*   inet:ipv6-address
|  |        +--ro unknowns
|  |        |  +--ro unknown* (type)
|  |        |     +--ro type     uint16
|  |        |     +--ro value?   yang:hex-string
|  |        +--ro multi-topology-ids
|  |        |  +--ro multi-topology-id*   uint16
|  |        +--ro node-flag-bits
|  |        |  +--ro flags?
|  |        |          bgp-ls-topo-t:node-flag-bits
|  |        +--ro opaque-node-attribute
|  |        |  +--ro opaque-attrs?   yang:hex-string
|  |        +--ro node-name
|  |        |  +--ro name?   string
|  |        +--ro sr-capabilities
|  |        |  +--ro srgb* (start-label)
|  |        |  |  +--ro start-label
|  |        |  |  |       ietf-rt-t:mpls-label
|  |        |  |  +--ro range-size?    uint32
|  |        |  +--ro isis-flags?   binary
|  |        +--ro sr-local-block
|  |        |  +--ro srlb* (start-label)
|  |        |  |  +--ro start-label
|  |        |  |  |       ietf-rt-t:mpls-label
|  |        |  |  +--ro range-size?    uint32
|  |        |  +--ro isis-flags?   binary
|  |        +--ro sr-algorithm
|  |        |  +--ro algorithm*   uint8
|  |        +--ro node-msd
|  |        |  +--ro msd* (msd-type)
|  |        |     +--ro msd-type     uint8
|  |        |     +--ro msd-value?   uint8
|  |        +--ro fads
|  |        |  +--ro fad* (algorithm)
|  |        |     +--ro algorithm              uint8
|  |        |     +--ro metric-type?           uint8
|  |        |     +--ro calc-type?             uint8
|  |        |     +--ro priority?              uint8
|  |        |     +--ro node-fad-attributes
|  |        |        +--ro node-fad-attribute* (type)
```

```
                |  |        |           +--ro type
                |  |        |           |       identityref
                |  |        |           +--ro exclude-any-affinity
                |  |        |           |  +--ro eag*   uint32
                |  |        |           +--ro include-any-affinity
                |  |        |           |  +--ro eag*   uint32
                |  |        |           +--ro include-all-affinity
                |  |        |           |  +--ro eag*   uint32
                |  |        |           +--ro flags
                |  |        |           |  +--ro flags?   binary
                |  |        |           +--ro exclude-srlg
                |  |        |           |  +--ro srlg*   uint32
                |  |        |           +--ro exclude-minimum-bw
                |  |        |           |  +--ro bw?
                |  |        |           |          ietf-rt-t:bandwidth-
ieee-float32
                |  |        |           +--ro exclude-maximum-delay
                |  |        |           |  +--ro delay?   uint32
                |  |        |           +--ro exclude-any-reverse-affin
ity
                |  |        |           |  +--ro eag*   uint32
                |  |        |           +--ro include-any-reverse-affin
ity
                |  |        |           |  +--ro eag*   uint32
                |  |        |           +--ro include-all-reverse-affin
ity
                |  |        |           |  +--ro eag*   uint32
                |  |        |           +--ro fa-unsupported-tlvs
                |  |        |              +--ro protocol-id?
                |  |        |              |       bgp-ls-topo-t:protoc
ol
                |  |        |              +--ro type*
                |  |        |                      uint16
                |  |        +--ro isis-area-identifiers
                |  |           +--ro area-address*
                |  |                   ietf-isis-t:area-address
                |  +--ro isis-node* (system-id psn-id as)
                |     +--ro system-id          ietf-isis-t:system-id
                |     +--ro psn-id             uint8
                |     +--ro as                 inet:as-number
                |     +--ro node-attributes
                |        +--ro node-attribute* (type)
                |           +--ro type
                |           |       identityref
                |           +--ro local-ipv4-router-ids
                |           |  +--ro router-id*   ietf-rt-t:router-id
                |           +--ro local-ipv6-router-ids
                |           |  +--ro router-id*   inet:ipv6-address
```

```
|           +--ro unknowns
|           |  +--ro unknown* (type)
|           |     +--ro type     uint16
|           |     +--ro value?   yang:hex-string
|           +--ro multi-topology-ids
|           |  +--ro multi-topology-id*   uint16
|           +--ro node-flag-bits
|           |  +--ro flags?
|           |          bgp-ls-topo-t:node-flag-bits
|           +--ro opaque-node-attribute
|           |  +--ro opaque-attrs?   yang:hex-string
|           +--ro node-name
|           |  +--ro name?   string
|           +--ro sr-capabilities
|           |  +--ro srgb* (start-label)
|           |  |  +--ro start-label
|           |  |  |       ietf-rt-t:mpls-label
|           |  |  +--ro range-size?    uint32
|           |  +--ro isis-flags?   binary
|           +--ro sr-local-block
|           |  +--ro srlb* (start-label)
|           |  |  +--ro start-label
|           |  |  |       ietf-rt-t:mpls-label
|           |  |  +--ro range-size?    uint32
|           |  +--ro isis-flags?   binary
|           +--ro sr-algorithm
|           |  +--ro algorithm*   uint8
|           +--ro node-msd
|           |  +--ro msd* (msd-type)
|           |     +--ro msd-type     uint8
|           |     +--ro msd-value?   uint8
|           +--ro fads
|           |  +--ro fad* (algorithm)
|           |     +--ro algorithm              uint8
|           |     +--ro metric-type?           uint8
|           |     +--ro calc-type?             uint8
|           |     +--ro priority?              uint8
|           |     +--ro node-fad-attributes
|           |        +--ro node-fad-attribute* (type)
|           |           +--ro type
|           |           |       identityref
|           |           +--ro exclude-any-affinity
|           |           |  +--ro eag*   uint32
|           |           +--ro include-any-affinity
|           |           |  +--ro eag*   uint32
|           |           +--ro include-all-affinity
|           |           |  +--ro eag*   uint32
|           |           +--ro flags
```

```
                |           |           |  +--ro flags?   binary
                |           |           +--ro exclude-srlg
                |           |           |  +--ro srlg*   uint32
                |           |           +--ro exclude-minimum-bw
                |           |           |  +--ro bw?
                |           |           |          ietf-rt-t:bandwidth-
 ieee-float32
                |           |           +--ro exclude-maximum-delay
                |           |           |  +--ro delay?   uint32
                |           |           +--ro exclude-any-reverse-affin
 ity
                |           |           |  +--ro eag*   uint32
                |           |           +--ro include-any-reverse-affin
 ity
                |           |           |  +--ro eag*   uint32
                |           |           +--ro include-all-reverse-affin
 ity
                |           |           |  +--ro eag*   uint32
                |           |           +--ro fa-unsupported-tlvs
                |           |              +--ro protocol-id?
                |           |              |       bgp-ls-topo-t:protoc
 ol
                |           |              +--ro type*
                |           |                      uint16
                |           +--ro isis-area-identifiers
                |              +--ro area-address*
                |                      ietf-isis-t:area-address
                +--ro unknowns
                   +--ro unknown* (nlri)
                      +--ro nlri          yang:hex-string
                      +--ro attributes?   yang:hex-string
  augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/bgp:bgp/bgp:neighbors
            /bgp:neighbor/bgp:afi-safis/bgp:afi-safi:
    +--rw link-state!
    |  +--rw prefix-limit
    |  |  +--rw max-prefixes?            uint32
    |  |  +--rw warning-threshold-pct?   rt-types:percentage
    |  |  +--rw teardown?                boolean
    |  |  +--rw idle-time?               union
    |  |  +--ro prefix-limit-exceeded?   boolean
    |  +--rw max-rate?       uint32
    |  +--rw max-number?     uint32
    +--rw link-state-spf
       +--rw prefix-limit
          |  +--rw max-prefixes?            uint32
          |  +--rw warning-threshold-pct?   rt-types:percentage
          |  +--rw teardown?                boolean
```

```
        |  +--rw idle-time?               union
        |  +--ro prefix-limit-exceeded?   boolean
        +--rw metric?         uint32
        +--rw status?         enumeration
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/bgp:bgp/bgp:neighbors
             /bgp:neighbor/bgp:statistics:
     +--ro updates-sent?             yang:zero-based-counter32
     +--ro updates-received?         yang:zero-based-counter32
     +--ro error-updates-received?   yang:zero-based-counter32
     +--ro computations?             yang:zero-based-counter32
     +--ro triggering-events?        yang:zero-based-counter32
     +---x clear {bt:clear-statistics}?
        +---w input
        |  +---w clear-at?   yang:date-and-time
        +--ro output
           +--ro clear-finished-at?   yang:date-and-time
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/bgp:bgp/bgp:peer-groups
             /bgp:peer-group/bgp:afi-safis/bgp:afi-safi:
     +--rw link-state!
     |  +--rw prefix-limit
     |  |  +--rw max-prefixes?            uint32
     |  |  +--rw warning-threshold-pct?   rt-types:percentage
     |  |  +--rw teardown?                boolean
     |  |  +--rw idle-time?               union
     |  |  +--ro prefix-limit-exceeded?   boolean
     |  +--rw max-rate?       uint32
     |  +--rw max-number?     uint32
     +--rw link-state-spf
        +--rw prefix-limit
        |  +--rw max-prefixes?            uint32
        |  +--rw warning-threshold-pct?   rt-types:percentage
        |  +--rw teardown?                boolean
        |  +--rw idle-time?               union
        |  +--ro prefix-limit-exceeded?   boolean
        +--rw metric?         uint32
        +--rw status?         enumeration
```

Figure 4: Complete tree diagram

Acknowledgements

TBA

Contributors

Thanks to all of the contributors.

Authors' Addresses

Mahesh Jethanandani (editor)
Arrcus, Inc
Email: mjethanandani@gmail.com

Keyur Patel
Arrcus, Inc
Email: keyur@arrcus.com

Aravind Babu MahendraBabu
Cisco, Inc
India
Email: aramahen@cisco.com