A YANG Model for BGP-LS, BGP-LS-VPN, and BGP-LS-SPF

draft-mahesh-lsvr-bgp-ls-yang-00

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 22 December 2022.

Copyright Notice

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

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

North-Bound Distribution of Link-State (LS) and Traffic Engineering (TE) Information Using BGP [RFC7752] 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 [I-D.ietf-lsvr-bgp-spf] 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.

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFI</td>
<td>Address Family Indicator</td>
</tr>
<tr>
<td>LS</td>
<td>Link-State</td>
</tr>
<tr>
<td>LSA</td>
<td>Link-State Advertisements</td>
</tr>
<tr>
<td>LSDB</td>
<td>Link-State Database</td>
</tr>
<tr>
<td>SAFI</td>
<td>Subsequent Address Family Indicator</td>
</tr>
<tr>
<td>SPF</td>
<td>Shortest Path First</td>
</tr>
<tr>
<td>TE</td>
<td>Traffic Engineering</td>
</tr>
</tbody>
</table>

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 enabled?        boolean
      |   +++-rw prefix-limit
      |      |   +++-rw max-prefixes?       uint32
      |      |   +++-rw shutdown-threshold-pct?  rt-types:percentage

Jethanandani & Patel    Expires 22 December 2022                [Page 3]

Internet-Draft      BGP LS, LS-VPN, LS-SPF YANG Model          June 2022

|   |   +++-rw restart-timer?    uint32
|   |   +++-rw max-rate?         uint32
|   |   +++-rw max-number?       uint32
|   |   +++-rw instance-id?      uint64
|   |   +++-rw asn-plus-bgp-ls-ids? uint32
|   |   +++-ro local-databases
|   |      |   +++-ro database* [vrf-name instance protocol-id area-id]
|   |   ...            
|   +++-rw link-state-spf
   |   +++-rw prefix-limit
   |      |   +++-rw max-prefixes?       uint32
   |      |   +++-rw shutdown-threshold-pct?  rt-types:percentage
   |      |   +++-rw restart-timer?      uint32
   |   +++-rw instance-identifier? uint64
   |   +++-rw algorithm-type?     spf-algorithm-type
   |   +++-rw node-status?        enumeration
   |   +++-ro log
   |      |   +++-ro event* [id]
   |      |   ...            
   |   +++-ro local-databases
   |      |   +++-ro database* [vrf-name instance protocol-id area-id]
   |      |     ...            

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 enabled?        boolean
      |   +++-rw prefix-limit
      |      |   +++-rw max-prefixes?       uint32
| +--rw shutdown-threshold-pct?   rt-types:percentage
| +--rw restart-timer?            uint32
| +--rw max-rate?                uint32
| +--rw max-number?              uint32
| +--rw instance-id?             uint64
| +--rw asn-plus-bgp-ls-ids?     uint32

---rw link-state-spf
  +--rw prefix-limit
    | +--rw max-prefixes?          uint32
    | +--rw shutdown-threshold-pct? rt-types:percentage
    | +--rw restart-timer?         uint32

---rw status?   enumeration

augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/bgp:bgp:bgp:global
  /bgp:afi-safis/bgp:afi-safi/bgp:statistics:
  ++--rw updates-sent?          yang:zero-based-counter32
  ++--rw updates-received?      yang:zero-based-counter32
  ++--rw local-ls-originated?   yang:zero-based-counter32

augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/bgp:bgp:bgp:neighbors
  /bgp:neighbor/bgp:statistics:
  ++--rw updates-sent?          yang:zero-based-counter32
  ++--rw updates-received?      yang:zero-based-counter32
  ++--rw error-updates-received? yang:zero-based-counter32
  ++--rw computations?           yang:zero-based-counter32
  ++--rw triggering-events?      yang:zero-based-counter32

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 enabled?              boolean
    | ++--rw prefix-limit
    |    | ++--rw max-prefixes?        uint32
    |    | ++--rw shutdown-threshold-pct? rt-types:percentage
    |    | ++--rw restart-timer?       uint32
    | ++--rw max-rate?             uint32
    | ++--rw max-number?           uint32
    | ++--rw instance-id?          uint64
    | ++--rw asn-plus-bgp-ls-ids?  uint32
  ++--rw link-state-spf
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 to add statistics related to LS, and at a neighbor level to add support for the three new AFI/SAFI.

The model imports Common YANG Data Types [RFC6991], A YANG Data Model for Routing Management (NMIA Version) [RFC8349], and BGP Model for Service Provider Network [I-D.ietf-idr-bgp-model].
(NMDA Version)."

import ietf-bgp {
    prefix bgp;
    reference
        "I-D.ietf-idr-bgp-model: BGP YANG Model for Service Provider Networks.";
}

import ietf-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:  <http://tools.ietf.org/wg/lsvr>
     WG List:  <lsvr@ietf.org>

    Authors: Mahesh Jethanandani (mjethanandani at gmail.com),
             Keyur Patel (keyur at arrcus.com)";

description
    "This module contains contains management information for BGP-LS database."
revision 2022-06-21 {
  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 7752: Link-State Info Distribution using BGP.";
}

identity bgp-ls-vpn {
  base bt:afi-safi-type;
  description
    "BGP Link-State VPN.";
  reference
    "RFC 7752: Link-State Info Distribution using BGP.";
}

identity bgp-ls-spf {
  base bt:afi-safi-type;
  description
    "BGP Link-State Shortest Path First (BGP-LS-SPF).";
  reference
}
typedef spf-algorithm-type {
    type enumeration {
        enum algorithm-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 algorithm-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 spf-algorithm-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 spf-status-reachable {
        description
            "The link is reachable in the current SPF topology."
    }
    enum spf-status-unreachable {
        description
            "The link is unreachable in the current SPF topology."
    }
}
default spf-status-reachable;
description
    "Sets SPF-Status of the corresponding LS Link NLRI."

}

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

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

    }

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

    leaf enabled {
        type boolean;
        default false;
        description
            "Enable/disable receiving/sending of Link-State NLRI(s)."
    }

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

    leaf max-rate {
        type uint32;
        units per-second;
        default 200;
Maximum rate at which Link-State NLRIs will be advertised or withdrawn from neighbors.

RFC 7752: Link-State Info Distribution Using BGP.

leaf max-number {
  type uint32;
  description "Maximim number of Link-State NLRIs stored in a router's RIB."
  reference "RFC 7752: Link-State Info Distribution Using BGP."
}

leaf instance-id {
  type uint64;
  description "64-bit Instance-ID."
  reference "RFC 7752: Link-State Info Distribution Using BGP."
}

leaf asn-plus-bgp-ls-ids {
  type uint32;
  description "A pair of ASN and BGP-LS identifiers per flooding set in which the node participates."
  reference "RFC 7752: Link-State Info Distribution 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-spfo' " {
  description
    "Include this container for BGP Link-State SPF specific configuration";
} description "BGP Linkstate-SPF configuration options";

  uses bgp:mp-all-afi-safi-common;
}
leaf algorithm-type {
    type spf-algorithm-type;
    default algorithm-normal;
    description
        "SPF Algorithm type associated with Link-State AFI SAFI";
}

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

augment "/rt:routing/rt:control-plane-protocols" +
    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" +
    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" { 
  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 7752: Link-State Info Distribution Using BGP.";
  }

  leaf updates-received {
    type yang:zero-based-counter32;
    description
      "Total number of Link-State NLRI updates received.";
    reference
      "RFC 7752: Link-State Info Distribution Using BGP.";
  }

  leaf local-ls-originated {
    type yang:zero-based-counter32;
    description
      "Total number of locally originated Link-State NLRIs.";
    reference
      "RFC 7752: Link-State Info Distribution Using BGP.";
  }
}

augment "/rt:routing/rt:control-plane-protocols" + 
"/rt:control-plane-protocol/bgp:bgp/bgp:neighbors" + 
"/bgp:neighbor/bgp: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 7752": Link-State Info Distribution Using BGP."
]
leaf updates-received {
    type yang:zero-based-counter32;
    description
        "Total number of Link-State NLRIs updates received per
         neighbor.";
    reference
        "RFC 7752": Link-State Info Distribution 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 7752": Link-State Info Distribution Using BGP.";
}
leaf computations {
    type yang:zero-based-counter32;
    description
        "Count of number of SPF computations made.";
    reference
        "I-D.ietf-lsvr-bgp-spf: BGP Link-State SPF Routing.";
}
leaf triggering-events {
    type yang:zero-based-counter32;
    description
        "SPF triggering events.";
    reference
        "I-D.ietf-lsvr-bgp-spf: BGP Link-State SPF Routing.";
}
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;
}

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 [RFC6991]

<CODE BEGINS> file "ietf-bgp-lsdb@2022-06-21.yang"

module ietf-bgp-lsdb {
    yang-version 1.1;
    prefix bgp-lsdb;

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

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

    organization
    "IETF LSVR Working Group";
    contact
    "WG Web:  <http://tools.ietf.org/wg/lsvr>
    WG List:  <lsvr@ietf.org>"
Authors: Mahesh Jethanandani (mjethanandani at gmail.com), Keyur Patel (keyur at arrcus.com)

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

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

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

This version of this YANG module is part of RFC XXXX (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself for full legal notices.

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

revision 2022-06-21 {
  description
    "Initial Version";
  reference
    "RFC XXXX, BGP Model for Link State Distribution.";
}

// Typedefs
typedef lsdb-proto-id-type {
  type enumeration {
    enum unknown {
      description "Unknown protocol.";
    }
    enum isis-level1 {
typedef lsdb-attr-type {
    type enumeration {
        enum unknown {
            description "Unknown attribute.";
        }
        enum node-pdu-auth-info {
            description "Node authentication information attribute.";
        }
        enum node-hostname {
            description "Node Hostname attribute.";
        }
        enum node-flags {
            description "Node Flags attribute.";
        }
        enum node-flags {
            description "Node Flags attribute.";
        }
    }
}

description "LS database proto id type.";

description "OSPF Version 3.";

description "BGP.";

description "IS-IS Level 1.";

description "IS-IS Level 2.";

description "OSP of Version 2.";

description "Direct.";

description "Static";

description "OSPF Version 3.";

description "BGP.";

description "LS database proto id type.";

type enumeration {
    enum unknown {
        description "Unknown attribute.";
    }
    enum node-pdu-auth-info {
        description "Node authentication information attribute.";
    }
    enum node-hostname {
        description "Node Hostname attribute.";
    }
    enum node-flags {
        description "Node Flags attribute.";
    }
}
enum node-opaque {
    description
    "Node Opaque attribute.";
}

enum node-as {
    description
    "Node AS Number attribute.";
}

enum node-isis-node-id {
    description
    "Node ISIS Node-Id attribute.";
}

enum node-ipv4-rtr-id {
    description
    "Node IPv4 Router-Id attribute.";
}

enum node-ipv6-rtr-id {
    description
    "Node IPv6 Router-Id attribute.";
}

enum node-sid-index {
    description
    "Node SID Index attribute.";
}

enum node-topo-ids {
    description
    "Node Topology-IDs attribute.";
}

enum node-isis-nlpid {
    description
    "Node NLPID attribute.";
}

enum node-isis-area-id {
    description
    "Node ISIS Area-Id attribute.";
}

enum node-ospf-area-id {
    description
    "Node OSPF Area-Id attribute.";
} enum node-admin-tag {
    description
    "Node Admin-Tags attribute.";
}
} enum node-srgb-range {
    description
    "Node SRGB Range attribute.";
}
} enum node-spf-status {
    description
    "Node SPF Status attribute.";
}
} enum node-spf-algorithm {
    description
    "Node SPF Algorithm Type attribute.";
}
} enum link-name {
    description
    "Link Name attribute.";
}
} enum link-proto-id {
    description
    "Link Protocol-Id attribute.";
}
} enum link-local-id {
    description
    "Link Local-Id attribute.";
}
}
} enum link-remote-id {
    description
    "Node Remote-Id attribute.";
}
} enum link-local-isis-node-id {
    description
    "Link Local ISO Node-Id attribute.";
}
} enum link-remote-isis-node-id {
    description
    "Link Remote ISO Node-Id attribute.";
}
}
enum link-local-ipv4-rtr-id {
  description
  "Link Local IPv4 Router-Id attribute.";
}

enum link-remote-ipv4-rtr-id {
  description
  "Link Remote IPv4 Router-Id attribute.";
}

enum link-local-ipv6-rtr-id {
  description
  "Link Local IPv6 Router-Id attribute.";
}

enum link-remote-ipv6-rtr-id {
  description
  "Link Remote IPv6 Router-Id attribute.";
}

enum link-circuit-id {
  description
  "Link Local Circuit-Id attribute.";
}

enum link-igp-metric {
  description
  "Link IGP metric attribute.";
}

enum link-mtu {
  description
  "Link MTU attribute.";
}

enum link-max-bandwidth {
  description
  "Link Maximum Bandwidth attribute.";
}

enum link-max-reserve-bandwidth {
  description
  "Link Maximum Reserved Bandwidth attribute.";
}

enum link-unreserve-bandwidth {
  description
  "Link Unreserved Bandwidth attribute.";
}

enum link-default-te-metric {
description
   "Link TE-Default Metric attribute."
}
enum link-protection-type {
   description
   "Link Protection-Type attribute."
}
enum link-opaque {
   description
   "Link Opaque attribute."
}
enum link-mpls-proto-mask {
   description
   "Link MPLS-Protocol-Mask attribute."
}
enum LSDB_LINK_LOCAL_IPV4_ADDR {
   description "Link Local IPv4 Address attribute."
}
enum LSDB_LINK_LOCAL_IPV6_ADDR {
   description "Link Local IPv6 Address attribute."
}
enum LSDB_LINK_REMOTE_IPV4_ADDR {
   description "Link Remote IPv4 Address attribute."
}
enum LSDB_LINK_REMOTE_IPV6_ADDR {
   description "Link Remote IPv6 Address attribute."
}
enum LSDB_LINK_ISIS_LEVEL {
   description "Link ISIS Level attribute."
}
enum LSDB_LINK_ADMIN_TAG {
   description "Link Admin-Tag attribute."
}
enum LSDB_LINK_SRLG {
   description "Link SRLGs attribute."
}
enum LSDB_LINK_SPF_STATUS {
   description "Link SPF Status attribute."
}
enum LSDB_PREFIX_Iوغ_FLAGS {
   description "Prefix IGP-Flags attribute."
}
enum LSDB_PREFIX_METRIC {

typedef lsdb-obj-type {
    type enumeration {
        enum LSDB_UNKNOWN_OBJ {
            description "Unknown attribute.";
        }
        enum LSDB_NODE_OBJ {
            description "Node Object.";
        }
        enum LSDB_LINK_OBJ {
            description "Link object.";
        }
        enum LSDB_PREFIX_OBJ {
            description "Prefix object.";
        }
    }
    description "Link-State database object type.";
}

// Groupings
grouping lsdb-obj-common {
    description "Common details for all object types.";
}
leaf handle {
  type uint64;
  description "Handle of the object."
}

leaf type {
  type lsdb-obj-type;
  description "Type of object."
}

leaf topology-id {
  type uint16;
  description "Id of the topology object belongs to."
}

leaf pdu-id {
  type uint32;
  description "Id of the original protocol PDU unit this object belongs to."
}

container attributes {
  description "List of attributes for a given object."
}

list attribute {
  key "attribute-handle attribute-type"
}

  leaf attribute-handle {
    type uint64;
    description "Handle of the attribute."
  }

  leaf attribute-type {
    type lsdb-attr-type;
    description "Type of attribute."
  }

  leaf attribute-length {
type uint16;
description
  "Length of attribute value in bytes."
};

leaf attribute-value {

type union {
  type uint8;
  type uint16;
  type uint32;
  type uint64;
  type boolean;
  type string;
  type inet:ip-address;
  type yang:hex-string;
}
description
  "Actual value of the attribute."
};
description
  "Details of a single attribute."
;
}

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

container local-databases {
  config false;
  description
    "List of local databases.";

  list database {
    key "vrf-name instance protocol-id area-id";
    description
      "Operational state for a given database."
    }

  leaf vrf-name {

leaf instance {
    type string;
    description
        "Name of the application instance this database belongs to.";
}

leaf protocol-id {
    type lsdb-proto-id-type;
    description
        "Id of the application protocol this database belongs to.";
}

leaf area-id {
    type uint32;
    description
        "Id of the protocol area this database belongs to.";
}

container nodes {
    description
        "List of node objects for a given database.";
    list node {
        key "handle";
        leaf node-id {
            type string;
            description
                "Id of the node object.";
        }
        uses lsdb-obj-common;
    }
}
"List of link objects for a given node object."

```yaml
list link {
key "handle";

leaf local-id {
  type uint32;
  description
    "Local identifier of the link object."
}

leaf local-address {
  type string;
  description
    "Local address of the link object."
}

leaf local-node-id {
  type uint32;
  description
    "Local Node identifier of the link object."
}

leaf remote-id {
  type uint32;
  description
    "Remote identifier of the link object."
}

leaf remote-address {
  type string;
  description
    "Remote address of the link object."
}

leaf remote-node-id {
  type uint32;
  description
    "Remote Node identifier of the link object."
}
```
uses lsdb-obj-common;
description
  "Details of a single link object."
};
}
}

container prefixes {
  description
    "List of prefix objects for a given node object."
  }
  list prefix {
    key "handle"
    leaf prefix-key {
      type string;
      description
        "Key value for the prefix object."
    }
  }
}

uses lsdb-obj-common;
description
  "Details of a single prefix object."
};
}

description
  "Details of a single node object."
};
This document registers the following YANG modules in the "YANG Module Names" registry [RFC6020]:

Name: ietf-bgp-ls
Prefix: bgp-ls
Reference: RFC XXXX

Name: 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


7.2. Informative References

Appendix A. Appendix 1 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 enabled?          boolean
    |  +--rw prefix-limit
    |     |  +--rw max-prefixes?    uint32
    |     |  +--rw shutdown-threshold-pct? rt-types:percentage
    |     |  +--rw restart-timer?    uint32
    +--rw max-rate?           uint32
    +--rw max-number?         uint32
    +--rw instance-id?        uint64
    +--rw asn-plus-bgp-ls-ids? uint32
    +--ro local-databases
      +--ro database* [vrf-name instance protocol-id area-id]
        +--ro vrf-name     string
        +--ro instance     string
        +--ro protocol-id  lsdb-proto-id-type
        +--ro area-id      uint32
        +--ro nodes
---ro node* [handle]
|   ---ro node-id?        string
|   ---ro handle          uint64
|   ---ro type?           lsdb-obj-type
|   ---ro topology-id?    uint16

---ro pdu-id?        uint32
---ro attributes
   ---ro attribute*
      [attribute-handle attribute-type]
      ---ro attribute-handle    uint64
      ---ro attribute-type      lsdb-attr-type
      ---ro attribute-length?   uint16
      ---ro attribute-value?    union

---ro links
   ---ro link* [handle]
      ---ro local-id?        uint32
      ---ro local-address?   string
      ---ro local-node-id?   uint32
      ---ro remote-id?       uint32
      ---ro remote-address?  string
      ---ro remote-node-id?  uint32
      ---ro handle           uint64
      ---ro type?            lsdb-obj-type
      ---ro topology-id?     uint16
      ---ro pdu-id?          uint32
      ---ro attributes
         ---ro attribute*
            [attribute-handle attribute-type]
            ---ro attribute-handle    uint64
            ---ro attribute-type      lsdb-attr-type
            ---ro attribute-length?   uint16
            ---ro attribute-value?    union

---ro prefixes
   ---ro prefix* [handle]
      ---ro prefix-key?       string
      ---ro handle            uint64
      ---ro type?             lsdb-obj-type
      ---ro topology-id?      uint16
      ---ro pdu-id?           uint32
      ---ro attributes
++--ro attribute*
      [attribute-handle attribute-type]
      ++--ro attribute-handle    uint64
      ++--ro attribute-type
      |       lsdb-attr-type
      ++--ro attribute-length?   uint16
      ++--ro attribute-value?    union

++--rw link-state-spf
++--rw prefix-limit
    | ++--rw max-prefixes?         uint32
    | ++--rw shutdown-threshold-pct? rt-types:percentage
    | ++--rw restart-timer?        uint32
/*
   +++ro attribute-handle      uint64
   +++ro attribute-type
   |   lsdb-attr-type
   +++ro attribute-length?     uint16
   +++ro attribute-value?      union

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 enabled?          boolean
   |   +++rw prefix-limit
   |   |   +++rw max-prefixes?   uint32
   |   |   +++rw shutdown-threshold-pct?  rt-types:percentage
   |   |   +++rw restart-timer?   uint32
   |   +++rw max-rate?         uint32
   |   +++rw max-number?       uint32
   |   +++rw instance-id?      uint64
   |   +++rw asn-plus-bgp-ls-ids?  uint32
   +++rw link-state-spf
   |   +++rw prefix-limit
   |   |   +++rw max-prefixes?   uint32
   |   |   +++rw shutdown-threshold-pct?  rt-types:percentage
   |   |   +++rw restart-timer?   uint32
   +++rw metric?              uint32
   +++rw status?              enumeration

augment /rt:routing/rt:control-plane-protocols
   /rt:control-plane-protocol/bgp:bgp/bgp:global
   /bgp:afi-safis/bgp:afi-safi/bgp:statistics:
   +++ro updates-sent?      yang:zero-based-counter32
   +++ro updates-received?  yang:zero-based-counter32
   +++ro local-ls-originated? yang:zero-based-counter32

augment /rt:routing/rt:control-plane-protocols
   /rt:control-plane-protocol/bgp:bgp/bgp:neighbors
   /bgp:neighbor/bgp:statistics:
   +++rw updates-sent?      yang:zero-based-counter32
   +++rw updates-received?  yang:zero-based-counter32
   +++rw error-updates-received? yang:zero-based-counter32
   +++rw computations?      yang:zero-based-counter32
   +++rw triggering-events? yang:zero-based-counter32

augment /rt:routing/rt:control-plane-protocols
   /rt:control-plane-protocol/bgp:bgp/bgp:peer-groups
*/
 Acknowledgements

TBA

Contributors

Thanks to all of the contributors.

Authors' Addresses

Mahesh Jethanandani (editor)
Arrcus, Inc

Jethanandani & Patel Expires 22 December 2022

Internet-Draft BGP LS, LS-VPN, LS-SPF YANG Model June 2022

Email: mjethanandani@gmail.com

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