

Spring Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 28, 2018

D. Jain
K. Raza
Cisco
B. Decraene
Orange
Z. Jiang
Tencent
Jun 26, 2018

**YANG data model for BGP Segment Routing Extensions
draft-dhjain-spring-bgp-sr-yang-00.txt**

Abstract

This document defines a YANG data model that can be used to configure and manage Segment Routing extensions in BGP.

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 December 28, 2018.

Copyright Notice

Copyright (c) 2018 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 Simplified BSD License text as described in [Section 4](#).e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1.	Introduction	2
1.1.	Requirements Language	3
2.	BGP Segment Routing Yang model	3
2.1.	Overview	3
2.2.	SR Prefix SID (SR MPLS)	4
2.3.	Egress Peer Engineering	4
2.4.	SR Policy	5
2.5.	Automatic Steering	5
2.6.	SRv6 SIDs	6
3.	Yang Tree	6
3.1.	SR Prefix Sid (SR MPLS)	6
3.2.	Egress Peer Engineering	7
3.3.	SR Policy	8
3.4.	Automatic Steering	10
3.5.	SRv6 SIDs	11
4.	Yang Module	12
5.	IANA Considerations	30
6.	Security Considerations	30
7.	Acknowledgements	30
8.	References	30
8.1.	Normative References	30
8.2.	Informative References	32
	Authors' Addresses	33

[1.](#) Introduction

YANG [[RFC6020](#)] is a data definition language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [[RFC6241](#)]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces (e.g. ReST) [[RFC8040](#)] and encodings other than XML (e.g. JSON)

[RFC7951] are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interfaces, such as CLI and programmatic APIs.

This document defines the YANG model for Segment Routing specific extensions in BGP.

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. BGP Segment Routing Yang model

2.1. Overview

Segment Routing (SR), as defined in [\[I-D.ietf-spring-segment-routing\]](#), leverages the source routing paradigm where a node steers a packet through an ordered list of instructions, called segments. SR, thus, allows enforcing a flow through any topological path and/or service chain while maintaining per-flow state only at the ingress nodes to the SR domain.

When applied to ipv6 data-plane (i.e. SRv6), the ordered set of instructions are realized via SRv6 SIDs. The various functions and behaviors corresponding to network programming using SRv6 are specified in [\[I-D.filsfils-spring-srv6-network-programming\]](#).

This document defines Yang model for the Segment Routing extensions applicable for BGP as following:

- o Prefix sid extensions in the context of SR MPLS, as described in [\[I-D.ietf-idr-bgp-prefix-sid\]](#).
- o Egress Peer Engineering (EPE) as described in [\[I-D.ietf-spring-segment-routing-central-epe\]](#).
- o BGP signaled SR Policy as described in [\[I-D.ietf-idr-segment-routing-te-policy\]](#).
- o Automatic Steering as described in [\[I-D.ietf-spring-segment-routing-policy\]](#) and [\[I-D.ietf-idr-segment-routing-te-policy\]](#).

- o SRv6 VPN extensions as described in [I-D.[draft-dawra-idr-srv6-vpn](#)].

The Yang extensions proposed in this model augment the base BGP model defined in [I-D.[ietf-idr-bgp-model](#)].

Note: Base BGP model does not have a common structure for BGP RIB. The placeholder containers defined in this model can be removed once base BGP model has the BGP RIB structure.

The modeling in this document complies with the Network Management Datastore Architecture (NMDA) [RFC8342]. The operational state data is combined with the associated configuration data in the same hierarchy [I-D.[ietf-netmod-rfc6087bis](#)]. When protocol states are retrieved from the NMDA operational state datastore, the returned states cover all "config true" (rw) and "config false" (ro) nodes defined in the schema.

[2.2.](#) SR Prefix SID (SR MPLS)

Prefix SID attribute in BGP in the context of SR MPLS, carries the label index and SRGB block information.

- o The configuration to attach the label index is modeled as a new route-policy set action. BGP policy actions from the BGP policy module defined in base BGP yang model [I-D.[ietf-idr-bgp-model](#)] are augmented for this purpose.
- o The configuration related to SR Mapping Server in the context of BGP prefix SID, is TBD.
- o Prefix SID attribute received with the BGP route is modeled under BGP AF mode for select address families. This information is applicable per route.

[2.3.](#) Egress Peer Engineering

Egress Peer Engineering (EPE) in the context of Segment Routing is described in [I-D.[ietf-spring-segment-routing-central-epe](#)]. EPE is enabled in the context of BGP neighbor session. Three different types of EPE SIDs namely, Peer node SID, Peer adjacency SID and Peer set SID correspond to the segments required for source routed inter domain paths. EPE SID(s) for each type above, can be statically configured or dynamically allocated by the node. Further, FRR backup policy and backup SID can be specified per EPE. The configuration and state for the EPE parameters is modeled by augmenting the neighbor container defined in the base BGP model

[[I-D.ietf-idr-bgp-model](#)]. The configuration and the applied config state are applicable for neighbor-groups as well.

2.4. SR Policy

Architecture for SR Policies is described in [[I-D.ietf-spring-segment-routing-policy](#)]. BGP Signaled SR Policies are described in the [[I-D.ietf-idr-segment-routing-te-policy](#)]. Following Yang extensions for SR Policy configuration and state data are applicable:

- o Addition of identities extending the BGP-AFI-SAFI base identity. This is to add two new address families namely IPv4 SR-policy and IPv6 SR-policy, as described in [[I-D.ietf-idr-segment-routing-te-policy](#)].
- o BGP Signaled SR Policy candidate paths. These refer to the explicit candidate paths signaled via BGP as SAFI NLRIs, state of which is applicable in the context of BGP speaker process. This is modeled by adding SR Policy address family specific container under generic BGP afi-safi list entry defined in the base BGP model [[I-D.ietf-idr-bgp-model](#)].
- o On Demand SR Policy candidate paths. These refer to the dynamic candidate paths as described in [[I-D.ietf-spring-segment-routing-policy](#)]. There are two parts to this in the context of BGP. A set of authorized SR Policy colors for on demand policy triggers, and the actual instantiated candidate paths per BGP next-hop. New containers and lists are added under BGP global mode to model this information.
- o SR Policy state in the context of BGP speaker. This represents the state SR Policies (regardless of method of instantiation per candidate path). The SR Policy state is maintained in the context of BGP speaker process to realize the Automatic Steering of overlay routes. Automatic Steering extensions are described in the next section.

Note: The common parameters and datatypes for the SR Policy, currently defined in this model, should be imported from SR Policy Manager model, once available.

2.5. Automatic Steering

Automatic Steering (AS) refers to the ability to forward traffic over a SR Policy on the head-end, as described in [[I-D.ietf-spring-segment-routing-policy](#)]. When a BGP route is received with the color extended community and if the color value

matches the color of an authorized SR Policy installed on the head-end, the route is programmed to resolve over SR Policy in forwarding. Automatic Steering information associated with the BGP routes is modeled as state information per route.

TBD: The configuration parameters for Automatic Steering are yet to be added as an augmentation to the BGP route policy model. Such as, extensions for opaque color extended community in BGP policy model, and the Color Only (CO) flags controlling the Automatic Steering behavior as described in [[I-D.ietf-idr-segment-routing-te-policy](#)].

2.6. SRv6 SIDs

SRv6 extensions defined here are correspond to the VPN programming via SRv6 as described in [[I-D.draft-dawra-idr-srv6-vpn](#)]. SRv6 sid allocation mode is applicable in the context of ipv4 unicast and ipv6 unicast SAFI under VPN context. This is modeled by adding new containers under the respective AFI/SAFIs from the base BGP model [[I-D.ietf-idr-bgp-model](#)].

The common data types for SRv6 are imported form [[I-D.draft-raza-spring-srv6-yang](#)].

TBD: Base BGP model [[I-D.ietf-idr-bgp-model](#)], in its current form is not scoped within the context of a Network Instance. Therefore, the context of a VRF is not fully realized. The extensions done in this model should fall within the scope of a VRF, once the top BGP container is linked under Network Instance.

3. Yang Tree

3.1. SR Prefix Sid (SR MPLS)


```

module: ietf-bgp-sr
augment /rpol:routing-policy/rpol:policy-definitions/rpol:policy-definition/
rpol:statements/rpol:statement/rpol:actions/bgp-pol:bgp-actions:
  +--rw set-label-index?   uint32

augment /bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv4-labeled-
unicast:
  +--ro routes
    +--ro route* [prefix neighbor add-path-id]
      +--ro prefix          inet:ip-prefix
      +--ro neighbor        inet:ip-address
      +--ro add-path-id     uint32
      +--ro prefix-sid
      | +--ro label-index?   uint32
      | +--ro originator-srgb
      |   +--ro srgb-ranges* [srgb-min srgb-max]
      |     +--ro srgb-min   rt-types:mpls-label
      |     +--ro srgb-max   rt-types:mpls-label
    augment /bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv6-labeled-
    unicast:
      +--ro routes
        +--ro route* [prefix neighbor add-path-id]
          +--ro prefix          inet:ip-prefix
          +--ro neighbor        inet:ip-address
          +--ro add-path-id     uint32
          +--ro prefix-sid
          | +--ro label-index?   uint32
          | +--ro originator-srgb
          |   +--ro srgb-ranges* [srgb-min srgb-max]
          |     +--ro srgb-min   rt-types:mpls-label
          |     +--ro srgb-max   rt-types:mpls-label
      ...

```

[3.2.](#) Egress Peer Engineering

Egress Peer Engineering Yang Tree applicable to neighbor and peer-group containers


```

module: ietf-bgp-sr

augment /bgp:bgp/bgp:neighbors/bgp:neighbor:
  +--rw egress-peer-engineering
    +--rw sid-allocation-type? enumeration
    +--rw explicit-sid? sid-type
    +--ro allocated-sid? sid-type
    +--rw peer-set-name? string
    +--rw backup
      | +--ro active? boolean
      | +--rw backup-type? enumeration
      | +--rw backup-peer? inet:ip-address
      | +--rw backup-sid? sid-type
    +--rw peer-adjacency* [first-hop-ipaddress]
      +--rw first-hop-ipaddress inet:ip-address
      +--ro first-hop-interface? string
      +--rw sid-allocation-type? enumeration
      +--rw explicit-sid? sid-type
      +--ro allocated-sid? sid-type
      +--rw backup
        +--ro active? boolean
        +--rw backup-type? enumeration
        +--rw backup-peer? inet:ip-address
        +--rw backup-sid? sid-type
  ...

```

3.3. SR Policy

On Demand Nexthop (ODN) policies triggered by BGP


```

augment /bgp:bgp/bgp:global:
  +--rw segment-routing
    +--rw on-demand-policies
      | +--ro authorized-colors
      | | +--ro colors* [color]
      | |   +--ro color      uint32
      | +--ro installed-policies
      |   +--ro sr-policy* [color end-point]
      |     +--ro color      uint32
      |     +--ro end-point  inet:ip-address
    +--ro policy-state
      +--ro sr-policy* [color end-point]
        +--ro color      uint32
        +--ro end-point  inet:ip-address
        +--ro policy-state? enumeration
        +--ro binding-sid?  sid-type
        +--ro steering-disabled? empty
        +--ro ref-count?   uint32

```

BGP Signaled Explicit SR Policies under ipv4 and ipv6 SR-Policy SAFI


```

augment /bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi:
  +--rw ipv4-srpolicy
    +--ro explicit-policies
      +--ro sr-policy* [distinguisher color endpoint]
        +--ro distinguisher          uint32
        +--ro color                  uint32
        +--ro endpoint               inet:ip-address
        +--ro preference?            uint32
        +--ro explicit-binding-sid
          | +--ro binding-sid?       sid-type
          | +--ro strict?            boolean
          | +--ro drop-on-invalid?   boolean
        +--ro usable?                boolean
        +--ro registered?            boolean

augment /bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi:
  +--rw ipv6-srpolicy
    +--ro explicit-policies
      +--ro sr-policy* [distinguisher color endpoint]
        +--ro distinguisher          uint32
        +--ro color                  uint32
        +--ro endpoint               inet:ip-address
        +--ro preference?            uint32
        +--ro explicit-binding-sid
          | +--ro binding-sid?       sid-type
          | +--ro strict?            boolean
          | +--ro drop-on-invalid?   boolean
        +--ro usable?                boolean
        +--ro registered?            boolean

```

3.4. Automatic Steering

Yang Tree for Automatic Steering with example of ipv4-unicast SAFI


```

module: ietf-bgp-sr
  augment /bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv4-unicast:
    +--ro routes
      +--ro route* [prefix neighbor add-path-id]
        +--ro prefix          union
        +--ro neighbor         inet:ip-address
        +--ro add-path-id      uint32
        +--ro automatic-steering
          | +--ro color?       -> /bgp:bgp/global/bgp-sr:segment-routing/
policy-state/sr-policy/color
          | +--ro end-point?   -> /bgp:bgp/global/bgp-sr:segment-routing/
policy-state/sr-policy/end-point
          | +--ro co-flag?     enumeration
          | +--ro binding-sid? -> /bgp:bgp/global/bgp-sr:segment-routing/
policy-state/sr-policy/binding-sid
    ...

```

[3.5.](#) SRv6 SIDs

SRv6 SID allocation mode in the context of select AFs

```

module: ietf-bgp-sr
  augment /bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv4-unicast:
    +--rw segment-routing
      +--rw srv6
        +--rw sid-alloc-mode? enumeration
  augment /bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv6-unicast:
    +--rw segment-routing
      +--rw srv6
        +--rw sid-alloc-mode? enumeration
    ...

```

SRv6 Received and local SID tree


```

augment /bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv4-unicast:
  +--ro routes
    +--ro route* [rd prefix neighbor add-path-id]
      +--ro rd          rt-types:route-distinguisher
      +--ro prefix      union
      +--ro neighbor    inet:ip-address
      +--ro add-path-id  uint32
      +--ro srv6
        +--ro received-sids* [received-sid]
          | +--ro received-sid  srv6-types:srv6-sid
        +--ro local-sids* [local-sid]
          +--ro local-sid  srv6-types:srv6-sid
          +--ro locator?   string
augment /bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv6-
unicast:
  +--ro routes
    +--ro route* [rd prefix neighbor add-path-id]
      +--ro rd          rt-types:route-distinguisher
      +--ro prefix      union
      +--ro neighbor    inet:ip-address
      +--ro add-path-id  uint32
      +--ro srv6
        +--ro received-sids* [received-sid]
          | +--ro received-sid  srv6-types:srv6-sid
        +--ro local-sids* [local-sid]
          +--ro local-sid  srv6-types:srv6-sid
          +--ro locator?   string

```

...

4. Yang Module

<CODE BEGINS> file "ietf-bgp-sr@2018-06-26.yang"

```

module ietf-bgp-sr {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-bgp-sr";
  // replace with IANA namespace when assigned
  prefix bgp-sr ;

  import ietf-routing-types {
    prefix rt-types;
  }

  import ietf-inet-types {
    prefix inet;
  }

```



```
import ietf-routing-policy {  
    prefix rpol;  
}
```

```
import ietf-bgp {  
    prefix bgp;  
}
```

```
import ietf-bgp-policy {  
    prefix bgp-pol;  
}
```

```
import ietf-bgp-types {  
    prefix bgp-types;  
}
```

```
import ietf-srv6-types {  
    prefix srv6-types;  
}
```

```
organization  
    "IETF Spring Working Group";
```

```
contact  
    "Spring working group - spring@ietf.org";
```

```
description  
    "This YANG module defines a data model to configure and  
    manage segment routing extensions in BGP.
```

Terms and Acronyms

AF : Address Family

BGP (bgp) : Border Gateway Protocol

EPE : Egress Peer Engineering

EVPN: Ethernet VPN

SR : Segment Routing

SID : Segment Identifier

SRv6 : Segment Routing with IPv6 Data plane

VPN : Virtual Private Network


```

    VRF : Virtual Routing and Forwarding

";

revision 2018-06-26 {
    description
        "Initial revision" ;
    reference "";
}

//
// New identities and typedefs for SR extensions
//

// SR Policy SAFI identities
identity IPV4_SRPOLICY {
    base bgp-types:AFI_SAFI_TYPE;
    description
        "IPv4 SR Policy (AFI,SAFI = 1,73)";
    reference "TBD";
}

identity IPV6_SRPOLICY {
    base bgp-types:AFI_SAFI_TYPE;
    description
        "IPv6 SR Policy (AFI,SAFI = 2,73)";
    reference "TBD";
}

// Sid type union
typedef sid-type {
    type union {
        type rt-types:mpls-label;
        type srv6-types:srv6-sid;
    }
    description "Type definition for Segment Identifier. This is
        a union type which can be either a SR MPLS SID in the
        form of a label, or a SRv6 SID in the form of
        an IPv6 address.";
    reference "TBD";
}

//
// SR Prefix SID related groupings
//

// Prefix SID attribute state in a route
grouping sr-route-prefix-sid {

```



```

description "SR Prefix SID attribute associated with BGP Route";
container prefix-sid {
  description "Prefix SID attribute";
  leaf label-index {
    type uint32;
    description "Label Index TLV carried with Prefix SID";
  }
  container originator-srgb {
    description "SRGB info of the originating node,
                 as signaled in the originator SRGB TLV";
    list srgb-ranges {
      key "srgb-min srgb-max";
      description "Concatenated ranges building the SRGB block";

      leaf srgb-min {
        type rt-types:mpls-label;
        description "Range min";
      }
      leaf srgb-max {
        type rt-types:mpls-label;
        description "Range max";
      }
    }
  }
}
}
}

//
// SR Egress Peer Engineering (EPE) related groupings
//
grouping epe-sid-alloc-mode {
  description "Common grouping for EPE mode and SID";
  leaf sid-allocation-type {
    type enumeration {
      enum EXPLICIT {
        description "EPE SID is configured";
      }
      enum DYNAMIC {
        description "EPE SID is generated by node";
      }
    }
  }
  default "DYNAMIC";
  description "SID allocation mode specifies whether the EPE SID
               is explicitly configured value, or a dynamically allocated
               value by the node. This applicable for EPE peer SID, EPE
peer
               adjacency SID and Peer set SID, depending on the context
               it is configured.";
}

```

}

Jain, et al.

Expires December 28, 2018

[Page 15]

```

    leaf explicit-sid {
        when "../mode = 'EXPLICIT'";
        type sid-type;
        description "Explicitly configured EPE SID value, when the sid-
allocation-type
                        is EXPLICIT";
    }

    leaf allocated-sid {
        type sid-type;
        config false;
        description "EPE SID value allocated by the node. When the sid allocation
type
                        is DYNAMIC, this would be a SID allocated by the node. In
the case
                        of EXPLICIT allocation type, this would typically be the
explicit sid
                        value configured by the user";
    }
}

grouping epe-backup-info {
    description "Parameters for EPE backup SID selection";
    container backup {
        description "Backup policy for this EPE";
        leaf active {
            type boolean;
            config false;
            description
                "Boolean indicating if the backup as per requested policy is
                active for this EPE. Typically when EPE Peer, Link or Set is
                down, backup SID as per backup policy, would become active";
        }
        leaf backup-type {
            type enumeration {
                enum PeerNodeSid {
                    description "Backup via another Peer Node SID to the
                        same AS. A Peer identifier is also
                        required when this backup-type is
                        selected";
                }
                enum PeerAdjSid {
                    description "Backup via remaining Peer Adjacencies to the
                        same peer";
                }
                enum PeerSetSid {
                    description "Backup via Remaining PeerNode SIDs in the
                        same PeerSet";
                }
            }
        }
    }
}

```

```
}  
enum IGP {  
    description "Pop the EPE SID and perform IP lookup";  
}
```

```

    }
    default "IGP";
    description "Type of the backup for this EPE";
    reference "TBD";
}

leaf backup-peer {
    when "../backup-type = 'PeerNodeSid'";
    type inet:ip-address;
    description "Peer identifier for the case when backup
                type is PeerNodeSid";
}

leaf backup-sid {
    type sid-type;
    description
        "Backup SID (of a EPE Peer, Peer Adjacency or Peer-Set) to be
        used as backup for this EPE";
}
}
}

grouping epe-config {
    description "Egress Peer Engineering (EPE) config grouping";
    container egress-peer-engineering {
        description "Egress Peer Engineering (EPE) config under BGP Peer";

        uses epe-sid-alloc-mode;

        leaf peer-set-name {
            type string;
            description "Make this EPE peer a member of the named
                        Peer Set.";
        }

        uses epe-backup-info;

        list peer-adjacency {
            key first-hop-ipaddress;
            description "EPE parameters for the adjacency links
                        over which multi-hop peering is setup";

            leaf first-hop-ipaddress {
                type inet:ip-address;
                description "First hop IP address of the link";
            }

            leaf first-hop-interface {

```



```

        type string;
        config false;
        description "The interface corresponding to the link";
    }

    uses epe-sid-alloc-mode;
    uses epe-backup-info;
}
}
}

//
// SR Policy Related Groupings
//
//Color and Endpoint of the SR Policy
grouping sr-policy-color-endpoint {
    description "Common grouping for SR Policy Color and
                Endpoint";
    leaf color {
        type uint32;
        description "Color of the policy";
    }

    leaf end-point {
        type inet:ip-address;
        description "Endpoint of the policy";
    }
}
// Authorized colors for On Demand SR Policy programming
grouping sr-odn-auth-colors {
    description
        "Authorized colors for On Demand (dynamic) SR Policies
        towards BGP nexthops";
    container authorized-colors {
        config false;
        description
            "Authorized colors for On Demand (dynamic) SR policies
            towards BGP nexthops";
        list colors {
            key "color";
            description "List of SR Policy Colors";
            leaf color {
                type uint32;
                description "Color value";
            }
        }
    }
}
}

```



```

grouping sr-policy-cmn-state {
  description "Common state parameters applicable to
              SR Policies";
  leaf policy-state {
    type enumeration {
      enum UP {
        description "SR Policy state UP";
      }
      enum DOWN {
        description "SR Policy state DOWN";
      }
    }
    description "SR Policy forwarding state";
  }

  leaf binding-sid {
    type sid-type;
    description "Binding SID of the SR Policy";
  }

  leaf steering-disabled {
    type empty;
    description "This attribute is set if steering
                is disabled on this SR policy";
  }

  leaf ref-count {
    type uint32;
    description "Count of routes steering over this policy";
  }
}

//
// SR Policy State grouping
//
grouping sr-policy-state {
  description "SR Policy State";
  container policy-state {
    config false;
    description "SR Policy State";
    list sr-policy {
      key "color end-point";
      description "List of SR Policies";

      uses sr-policy-color-endpoint;

      // State of the SR Policy in BGP

```



```

    uses sr-policy-cmn-state;
  }
}

grouping sr-exp-policy-cp-state {
  description "State of BGP signaled SR Policy (explicit)
    candidate paths";
  container explicit-policies {
    config false;
    description "BGP signaled explicit SR Policies";
    list sr-policy {
      key "distinguisher color end-point";
      description "List of BGP signaled explicit SR Policies";
      leaf distinguisher {
        type uint32;
        description "Distinguisher of the SR Policy
          candidate path";
      }

      uses sr-policy-color-endpoint;

      leaf preference {
        type uint32;
        description "Preference of the SR Policy candidate path";
      }
    }

    container explicit-binding-sid {
      description "Explicitly supplied Binding SID
        for this policy";
      leaf binding-sid {
        type sid-type;
        description "Binding SID value";
      }
      leaf strict {
        type boolean;
        description "Boolean indicating that the node
          must use only the supplied Binding SID
          for this SR Policy.
          reference: TBD";
      }
      leaf drop-on-invalid {
        type boolean;
        description "Boolean to indicate drop upon invalid
          policy, behavior. This overwrites the
          default behavior of fallback to IGP path
          , when SR Policy is (or becomes) invalid.
          reference: TBD";
      }
    }
  }
}

```



```

    }
  }

  leaf usable {
    type boolean;
    description "Boolean to indicate that the SR Policy is
                 usable on this node.
                 reference: TBD";
  }

  leaf registered {
    type boolean;
    description "Boolean to indicate that the SR policy
                 is registered with policy manager to
                 install the corresponding forwarding entry";
  }

  // TODO: Segment Lists and other parameters from SR Policy model
  //       to be imported here.
}
}

grouping sr-odn-policies {
  description "SR On Demand (dynamic) SR Policies";
  container installed-policies {
    config false;
    description "BGP triggered On Demand (dynamic) SR Policies
                 corresponding to the BGP nexthops";
    list sr-policy {
      key "color end-point";
      description "SR Policy list";
      uses sr-policy-color-endpoint;
    }
  }
}

grouping sr-policy-steering-state {
  description "Per route Automatic Steering parameters";
  container automatic-steering {
    description "Per route Automatic Steering parameters";
    leaf color {
      type leafref {
        path "/bgp:bgp/bgp:global/bgp-sr:segment-routing/" +
            "bgp-sr:policy-state/bgp-sr:sr-policy/" +
            "bgp-sr:color";
      }
    }
  }
}

```



```

        description "Color of the SR Policy being used for
                    Automatic Steering";
    }
    leaf end-point {
        type leafref {
            path "/bgp:bgp/bgp:global/bgp-sr:segment-routing/" +
                "bgp-sr:policy-state/bgp-sr:sr-policy/" +
                "bgp-sr:end-point";
        }
        description "End-point of the SR Policy being used
                    for Automatic Steering";
    }
    leaf co-flag {
        type enumeration {
            enum 00 {
                description "Color-Only flag 00";
            }
            enum 01 {
                description "Color-Only flag 01";
            }
            enum 10 {
                description "Color-Only flag 10";
            }
        }
        default "00";
        description "Color-Only (CO) flags applicable for
                    Automatic Steering of this route";
    }
    leaf binding-sid {
        type leafref {
            path "/bgp:bgp/bgp:global/bgp-sr:segment-routing/" +
                "bgp-sr:policy-state/bgp-sr:sr-policy/" +
                "bgp-sr:binding-sid";
        }
        description "Binding SID of the SR Policy";
    }
}

grouping route-key-leafs {
    description "Grouping for key leafs identifying a route";
    leaf prefix {
        type union {
            type inet:ip-prefix;
            type string;
        }
        description "BGP Prefix. This is a temp definition to
                    cover ip-prefix and other NLRI formats.

```



```

        Import the type once defined in base
        BGP RIB model";
    }
    leaf neighbor {
        type inet:ip-address;
        description "BGP Neighbor";
    }
    leaf add-path-id {
        type uint32;
        description "Add-path ID";
    }
}

grouping common-bgp-route-grouping {
    description "BGP route list" ;
    container routes {
        config false;
        description "BGP Route in local RIB";
        list route {
            key "prefix neighbor add-path-id";
            description "BGP route list";
            uses route-key-leafs;
        }
    }
}

grouping common-bgp-vpn-route-grouping {
    description "BGP route list" ;
    container routes {
        config false;
        description "BGP VPN Route in local RIB";
        list route {
            key "rd prefix neighbor add-path-id";
            description "Route List";

            leaf rd {
                type rt-types:route-distinguisher;
                description "Route Distinguisher";
            }
            uses route-key-leafs;
        }
    }
}

//
// SRv6 extensions related Groupings
//

```



```
// SRv6 VPN Sid allocation mode
grouping srv6-sid-mode {
  description "SRv6 VPN SID allocation mode";
  leaf sid-alloc-mode {
    type enumeration {
      enum per-ce {
        description "Allocate SRv6 SID per CE";
      }
      enum per-route {
        description "Allocate SRv6 SID per prefix";
      }
      enum per-vpn {
        description "Allocate SRv6 SID per VPN";
      }
    }
  }
  description "BGP SRv6 SID allocation model";
}
}
```

```
grouping srv6-attr-sid-info {
  description "SRv6 SID info per route";
  container srv6 {
    description "Per Route SRv6 parameters";
    list received-sids {
      key "received-sid";
      description "List of received SRv6 SIDs";
      leaf received-sid {
        type srv6-types:srv6-sid;
        description "Received SID";
      }
    }
    list local-sids {
      key "local-sid";
      description "List of local SRv6 SIDs";
      leaf local-sid {
        type srv6-types:srv6-sid;
        description "Local SID";
      }
      leaf locator {
        type string;
        description "SRv6 Locator";
      }
    }
  }
}
}
```

```
//
// BGP Specific Paramters
```



```

//
// Augment AF with route list
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
    "bgp:afi-safi/bgp:ipv4-unicast" {
    description
        "Augment BGP SAFI route";
    uses common-bgp-route-grouping;
}
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
    "bgp:afi-safi/bgp:ipv6-unicast" {
    description
        "Augment BGP SAFI route";
    uses common-bgp-route-grouping;
}
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
    "bgp:afi-safi/bgp:ipv4-labeled-unicast" {
    description
        "Augment BGP SAFI route";
    uses common-bgp-route-grouping;
}
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
    "bgp:afi-safi/bgp:ipv6-labeled-unicast" {
    description
        "Augment BGP SAFI route";
    uses common-bgp-route-grouping;
}
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
    "bgp:afi-safi/bgp:l3vpn-ipv4-unicast" {
    description
        "Augment BGP SAFI route";
    uses common-bgp-vpn-route-grouping;
}
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
    "bgp:afi-safi/bgp:l3vpn-ipv6-unicast" {
    description
        "Augment BGP SAFI route";
    uses common-bgp-vpn-route-grouping;
}
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
    "bgp:afi-safi/bgp:l2vpn-evpn" {
    description
        "Augment BGP SAFI route";
    uses common-bgp-vpn-route-grouping;
}

// SR Prefix SID Related.
// Prefix SID label index config via Route Policy

```



```

augment "/rpol:routing-policy/" +
    "rpol:policy-definitions/rpol:policy-definition/" +
    "rpol:statements/rpol:statement/" +
    "rpol:actions/bgp-pol:bgp-actions" {
    description
        "BGP policy actions to set label index";
    leaf set-label-index {
        type uint32;
        description "Label Index";
    }
}

// Prefix SID label in SAFI route
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
    "bgp:afi-safi/bgp:ipv4-labeled-unicast/bgp-sr:routes/bgp-sr:route" {
    description
        "Augment BGP AF Table for SR prefix sid Labels info";
    uses sr-route-prefix-sid;
}

augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
    "bgp:afi-safi/bgp:ipv6-labeled-unicast/bgp-sr:routes/bgp-sr:route" {
    description
        "Augment BGP AF Table for SR prefix sid Labels info";
    uses sr-route-prefix-sid;
}

// TBD: SR Mapping server related parameters.

// Egress Peer Engineering (EPE) related.
// EPE config under neighbor
augment "/bgp:bgp/bgp:neighbors/bgp:neighbor" {
    description
        "Egress Peer Engineering data";
    uses epe-config;
}

augment "/bgp:bgp/bgp:peer-groups/bgp:peer-group" {
    description
        "Egress Peer Engineering data";
    uses epe-config;
}

// SR Policy Related
// On Demand authorized colors table
// SR Policy state data
augment "/bgp:bgp/bgp:global" {
    description
        "Segment Routing parameters in BGP global model";
    container segment-routing {

```



```

    description "Segment Routing parameters";
    container on-demand-policies {
      description
        "Segment Routing On Demand Nexthop
        (ODN) SR Policies";
      uses sr-odn-auth-colors;
      uses sr-odn-policies;
    }
    uses sr-policy-state;
  }
}

// Steering state in overlay BGP routes
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
  "bgp:afi-safi/bgp:ipv4-unicast/bgp-sr:routes/bgp-sr:route" {
  description
    "Augment BGP SAFI route with steering info";
  uses sr-policy-steering-state;
}
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
  "bgp:afi-safi/bgp:ipv6-unicast/bgp-sr:routes/bgp-sr:route" {
  description
    "Augment BGP SAFI route with steering info";
  uses sr-policy-steering-state;
}
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
  "bgp:afi-safi/bgp:ipv4-labeled-unicast/bgp-sr:routes/bgp-sr:route" {
  description
    "Augment BGP SAFI route with steering info";
  uses sr-policy-steering-state;
}
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
  "bgp:afi-safi/bgp:ipv6-labeled-unicast/bgp-sr:routes/bgp-sr:route" {
  description
    "Augment BGP SAFI route with steering info";
  uses sr-policy-steering-state;
}
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
  "bgp:afi-safi/bgp:l3vpn-ipv4-unicast/bgp-sr:routes/bgp-sr:route" {
  description
    "Augment BGP SAFI route with steering info";
  uses sr-policy-steering-state;
}
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
  "bgp:afi-safi/bgp:l3vpn-ipv6-unicast/bgp-sr:routes/bgp-sr:route" {
  description
    "Augment BGP SAFI route with steering info";
  uses sr-policy-steering-state;
}

```



```

}
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
  "bgp:afi-safi/bgp:l2vpn-evpn/bgp-sr:routes/bgp-sr:route" {
  description
    "Augment BGP SAFI route with steering info";
  uses sr-policy-steering-state;
}

// BGP Signaled SR Policy explicit candidate paths state
augment "/bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi" {
  description "Augment IPv4 SR Policy SAFI list entry";
  container ipv4-srpolicy {
    when "../afi-safi-name = 'bgp-types:IPV4_SRPOLICY'" {
      description
        "Include this container for IPv4 SR Policy specific
        configuration";
    }
    description "IPv4 SR Policy specific parameters";
    uses sr-exp-policy-cp-state;
  }
}

augment "/bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi" {
  description "Augment IPv6 SR Policy SAFI list entry";
  container ipv6-srpolicy {
    when "../afi-safi-name = 'bgp-types:IPV6_SRPOLICY'" {
      description
        "Include this container for IPv6 SR Policy specific
        configuration";
    }
    description "IPv6 SR Policy specific parameters";
    uses sr-exp-policy-cp-state;
  }
}

// SRv6 VPN SID allocation mode configuration.
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
  "bgp:afi-safi/bgp:ipv4-unicast" {
  description
    "Augment BGP global IPv4 unicast AF mode
    to add SR specific parameters";
  container segment-routing {
    description "Segment Routing specific parameters";
    container srv6 {
      description "SRv6 specific parameters";
      uses srv6-sid-mode;
    }
  }
}

```



```

}

augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
  "bgp:afi-safi/bgp:ipv6-unicast" {
  description
    "Augment BGP global IPv6 unicast AF mode
    to add SR specific parameters";
  container segment-routing {
    description "Segment Routing specific parameters";
    container srv6 {
      description "SRv6 specific parameters";
      uses srv6-sid-mode;
    }
  }
}

// SRv6 local and remote sids per route.
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
  "bgp:afi-safi/bgp:ipv4-unicast/bgp-sr:routes/bgp-sr:route" {
  description
    "Augment AF route with SRv6 SID info";
  uses srv6-attr-sid-info;
}

augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
  "bgp:afi-safi/bgp:ipv6-unicast/bgp-sr:routes/bgp-sr:route" {
  description
    "Augment AF route with SRv6 SID info";
  uses srv6-attr-sid-info;
}

augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
  "bgp:afi-safi/bgp:l3vpn-ipv4-unicast/routes/route" {
  description
    "Augment AF route with SRv6 SID info";
  uses srv6-attr-sid-info;
}

augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
  "bgp:afi-safi/bgp:l3vpn-ipv6-unicast/bgp-sr:routes/bgp-sr:route" {
  description
    "Augment AF route with SRv6 SID info";
  uses srv6-attr-sid-info;
}

augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +
  "bgp:afi-safi/bgp:l2vpn-evpn/bgp-sr:routes/bgp-sr:route" {
  description
    "Augment AF route with SRv6 SID info";

```



```
    uses srv6-attr-sid-info;  
  }  
}
```

<CODE ENDS>

5. IANA Considerations

6. Security Considerations

The transport protocol used for sending the BGP Segment Routing data MUST support authentication and SHOULD support encryption. The data-model by itself does not create any security implications.

This draft does not change any underlying security issues inherent in [[I-D.ietf-idr-bgp-model](#)].

7. Acknowledgements

The authors would like to thank TBD for their detail reviews and comments.

8. References

8.1. Normative References

[I-D.dawra-idr-srv6-vpn]

Dawra, G., Filsfils, C., Dukes, D., Brissette, P., Camarillo, P., Leddy, J., daniel.voyer@bell.ca, d., daniel.bernier@bell.ca, d., Steinberg, D., Raszuk, R., Decraene, B., and S. Matsushima, "BGP Signaling of IPv6-Segment-Routing-based VPN Networks", [draft-dawra-idr-srv6-vpn-03](#) (work in progress), December 2017.

[I-D.filsfils-spring-srv6-network-programming]

Filsfils, C., Li, Z., Leddy, J., daniel.voyer@bell.ca, d., daniel.bernier@bell.ca, d., Steinberg, D., Raszuk, R., Matsushima, S., Lebrun, D., Decraene, B., Peirens, B., Salsano, S., Naik, G., Elmalky, H., Jonnalagadda, P., and M. Sharif, "SRv6 Network Programming", [draft-filsfils-spring-srv6-network-programming-04](#) (work in progress), March 2018.

[I-D.ietf-6man-segment-routing-header]

Previdi, S., Filsfils, C., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", [draft-ietf-6man-segment-routing-header-13](#) (work in progress), May 2018.

[I-D.ietf-idr-bgp-model]

Shaikh, A., Shakir, R., Patel, K., Hares, S., D'Souza, K., Bansal, D., Clemm, A., Zhdankin, A., Jethanandani, M., and X. Liu, "BGP Model for Service Provider Networks", [draft-ietf-idr-bgp-model-02](#) (work in progress), July 2016.

[I-D.ietf-idr-bgp-prefix-sid]

Previdi, S., Filsfils, C., Lindem, A., Sreekantiah, A., and H. Gredler, "Segment Routing Prefix SID extensions for BGP", [draft-ietf-idr-bgp-prefix-sid-18](#) (work in progress), March 2018.

[I-D.ietf-idr-segment-routing-te-policy]

Previdi, S., Filsfils, C., Jain, D., Mattes, P., Rosen, E., and S. Lin, "Advertising Segment Routing Policies in BGP", [draft-ietf-idr-segment-routing-te-policy-03](#) (work in progress), May 2018.

[I-D.ietf-netmod-rfc6087bis]

Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", [draft-ietf-netmod-rfc6087bis-20](#) (work in progress), March 2018.

[I-D.ietf-rtgwg-ni-model]

Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Network Instances", [draft-ietf-rtgwg-ni-model-04](#) (work in progress), September 2017.

[I-D.ietf-rtgwg-routing-types]

Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Routing Area Common YANG Data Types", [draft-ietf-rtgwg-routing-types-17](#) (work in progress), October 2017.

[I-D.ietf-spring-segment-routing]

Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", [draft-ietf-spring-segment-routing-15](#) (work in progress), January 2018.

- [I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., daniel.voyer@bell.ca, d., bogdanov@google.com, b., and P. Mattes, "Segment Routing Policy Architecture", [draft-ietf-spring-segment-routing-policy-01](#) (work in progress), June 2018.
- [I-D.raza-spring-srv6-yang]
Raza, K., Rajamanickam, J., Liu, X., Hu, Z., Hussain, I., Shah, H., daniel.voyer@bell.ca, d., Elmalky, H., Matsushima, S., Horiba, K., and A. Abdelsalam, "YANG Data Model for SRv6 Base and Static", [draft-raza-spring-srv6-yang-01](#) (work in progress), March 2018.
- [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>>.
- [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>>.
- [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>>.
- [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>>.

[8.2.](#) Informative References

- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", [RFC 7951](#), DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.
- [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>>.

Authors' Addresses

Dhanendra Jain
Cisco
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: dhjain@cisco.com

Kamran Raza
Cisco
2000 Innovation Drive
Kanata, ON K2K-3E8
CA

Email: skraza@cisco.com

Bruno Decraene
Orange
France

Email: bruno.decraene@orange.com

Zhichun Jiang
Tencent

Email: zcjiang@tencent.com

