Abstract

In this document we specify a new BGP capability - the Geo Coordinate Capability, and a new BGP attribute - the Geo Coordinate Attribute, for carrying the Geo Coordinate information in BGP.

1. Introduction

There are several potential applications as described hereby for the physical location information of BGP speakers [RFC4271] in a network.

In an "overlay network" without IGP s or where the "underlay network" belongs to a different administrative domain (e.g., using the BGP Tunnel Encapsulation Attribute [I-D.ietf-idr-tunnel-encaps]), the geographical location of the BGP speaker that sources the route in the network can be used to derive some rough sense of "distance" as a parameter in lieu of the IGP-metric in BGP path selection.

In the applications of "Service Function Chaining" [RFC7665], the Geo location information of the Service Function Forwarders or the Service Nodes, can help the design of Service Function Paths with better traffic pattern and a lower latency.

The knowledge of the physical location of BGP speakers can also be used to simplify the operational procedures for setting the outbound "MED" value in route advertisement.

In this document we specify a new BGP capability - the Geo Coordinate Capability, and a new BGP attribute - the Geo Coordinate Attribute, for carrying the Geo Coordinate information in BGP.

1.1. Specification of Requirements

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

The Geo Coordinate Capability is a new BGP capability [RFC5492]. The Capability Code for this capability is specified in the "IANA Considerations" section of this document. The Capability Length is 16 octets. The Capability Value consists of the following fields that specify the location of the speaker using the WGS-84 (World Geodetic System) reference coordinate system [WGS-84]:

<table>
<thead>
<tr>
<th>U</th>
<th>N</th>
<th>E</th>
<th>A</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td>Location Uncertainty</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Latitude Degr.</td>
<td>Latitude Milliseconds</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Longitude Degr.</td>
<td>Longitude Milliseconds</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Altitude</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

where:

U-bit: If the U-bit is set, it indicates that the "Location Uncertainty" field is specified. If the U-bit is clear, it indicates the "Location Uncertainty" field is unspecified.

N-bit: If the N-bit is set, it indicates the Latitude is north relative to the Equator. If the N-bit is clear, it indicates the Latitude is south of the Equator.

E-bit: If the E-bit is set, it indicates the Longitude is east of the Prime Meridian. If the E-bit is clear, it indicates the Longitude is West of the Prime Meridian.

A-bit: If the A-bit is set, it indicates the "Altitude" field is specified. If the A-bit is clear, it indicates the "Altitude" field is unspecified.

M-bit: If the M-bit is set, it indicates the "Altitude" is specified in meters. If the M-bit is clear, it indicates the "Altitude" is in centimeters.

Reserved: These bits are reserved. They SHOULD be set to zero by the sender and MUST be ignored by the receiver.

Location Uncertainty: Unsigned 16-bit integer indicating the number of centimeters of uncertainty for the location.
Latitude Degrees: Unsigned 8-bit integer with a range of 0 – 90 degrees north or south the Equator (northern or southern hemisphere, respectively).

Latitude Milliseconds: Unsigned 24-bit integer with a range of 0 – 3,599,999 (i.e., less than 60 minutes).

Longitude Degrees: Unsigned 8-bit integer with a range of 0 – 180 degrees east or west of the Prime Meridian.

Longitude Milliseconds: Unsigned 24-bit integer with a range of 0 – 3,599,999 (i.e., less than 60 minutes).

Altitude: Signed 32-bit integer containing the Height relative to the sea level in centimeters or meters. A negative height indicates that the location is below sea level.

3. The Geo Coordinate Attribute

The Geo Coordinate Attribute is an optional, transitive BGP attribute [RFC4271]. The type of the attribute is described in the IANA Considerations section, and the value of the attribute consists of one or more of the tuple encoded as shown below:

where the "AS number" and the "BGP Identifier" fields contain the AS number and the BGP Identifier [RFC4271, RFC6286] of the BGP speaker that sources or advertises the route, and the remaining fields specify the location of the speaker using the WGS-84 (World Geodetic System) reference coordinate system [WGS-84]. These location related fields are hereby given the same description as the ones in the "Geo
4. Operations

The Geo Coordinate Capability may be used by a BGP speaker to advertise its physical location to its neighbor. When an IGP (such as OSPF or ISIS) is involved and accessible, it could be advantageous for the Geo Coordinates to be carried in the IGP instead of in the OPEN for internal BGP ("IBGP") sessions.

When a BGP speakers receives the Geo Coordinate Capability in the OPEN message from a neighbor, it is up to the speaker and its local policy to decide how the information would be used.

The Geo Coordinate Attribute may be used by a BGP speaker to encode the physical location of the speaker in an UPDATE message. In the case that a route already contains the attribute, the speaker MAY prepend its AS number, its BGP Identifier, and the Geo coordinate information in the value field of the attribute, and adjust the attribute length accordingly. Depending on local policy, the speaker MAY also override the existing Geo Coordinate Attribute with its own information in route advertisement.

When a BGP speakers receives the Geo Coordinate Attribute in an UPDATE message from a neighbor, it is up to the speaker and the local policy to decide how this attribute would be handled and used.

The Geo Coordinate Capability in an OPEN message does not have any impact on how the Geo Coordinate Attribute in an UPDATE message (carried over the same session) would be handled.

5. Error Handling

The Geo Coordinate Attribute in an UPDATE message is considered malformed if the attribute length is not a non-zero multiple of 24.

An UPDATE message with a malformed Geo Coordinate Attribute SHALL be handled using the approach of "attribute discard" [RFC7606].
6. IANA Considerations

This document specifies a BGP capability, the Geo Coordinate Capability. The capability type needs to be allocated by IANA.

This document specifies a BGP attribute, the Geo Coordinate Attribute. The attribute type needs to be allocated by IANA.

7. Security Considerations

The underlying security issues for BGP are discussed in [RFC4271].

Since the Geo coordinates provide the exact location of the routing devices, disclosure may make the routing devices more susceptible to physical attacks. In situations where this is a concern (e.g., in military applications, or the topology of the network is considered proprietary information), the implementation MUST allow the Geo Location extension to be removed from the BGP’s OPEN and UPDATE messages.

8. Acknowledgments

The encoding of the Geo location is adapted from the "Geo Coordinate LISP Canonical Address Format" specified in the "LISP Canonical Address Format (LCAF)". We would like to thank the authors of that Document and particularly Dino Farinacci for subsequent discussions.

Thanks to Yi Yang for review and discussions of the Geo Coordinate encoding.

9. References

9.1. Normative References


[RFC5492] Scudder, J. and R. Chandra, "Capabilities Advertisement..."


9.2. Informative References


10. Authors’ Addresses

Enke Chen
Cisco Systems
170 W. Tasman Drive
San Jose, CA  95134
USA
Email: enkechen@cisco.com

Naiming Shen
Cisco Systems
170 W. Tasman Drive
San Jose, CA  95134
USA
Email: naiming@cisco.com
Abstract

Segment Routing (SR) allows for a flexible definition of end-to-end paths within IGP topologies by encoding paths as sequences of topological sub-paths, called "segments". These segments are advertised by the link-state routing protocols (IS-IS, OSPF and OSPFv3).

This draft defines extensions to the BGP Link-state address-family in order to carry segment information via BGP.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].
This Internet-Draft will expire on May 3, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction .................................................. 3
2. BGP-LS Extensions for Segment Routing ........................ 5
   2.1. Node Attributes TLVs .................................. 5
   2.1.1. SR-Capabilities TLV .............................. 5
   2.1.2. SR-Algorithm TLV ................................ 6
   2.1.3. SR Local Block TLV ................................ 7
   2.1.4. SRMS Preference TLV .............................. 7
   2.2. Link Attribute TLVs .................................. 8
   2.2.1. Adjacency SID TLV ................................ 9
   2.2.2. LAN Adjacency SID TLV ............................ 9
   2.3. Prefix Attribute TLVs ................................ 10
   2.3.1. Prefix-SID TLV .................................. 11
   2.3.2. IPv6 Prefix-SID TLV .............................. 12
   2.3.3. IGP Prefix Attributes TLV ........................ 13
   2.3.4. Source Router Identifier (Source Router-ID) TLV ... 14
   2.3.5. Range TLV ........................................ 14
   2.3.6. Binding SID TLV .................................. 15
   2.3.7. Binding SID SubTLVs ................................ 16
   2.4. Equivalent IS-IS Segment Routing TLVs/Sub-TLVs ........ 22
   2.5. Equivalent OSPF/OSPFv3 Segment Routing TLVs/Sub-TLVs .. 23
3. Procedures .................................................... 25
   3.1. Advertisement of a IS-IS Prefix SID TLV .............. 25
   3.2. Advertisement of a OSPF/OSPFv3 Prefix-SID TLV ........ 25
   3.3. Advertisement of a range of prefix-to-SID mappings in OSPF .................................. 26
   3.4. Advertisement of a range of IS-IS SR bindings ........ 26
   3.5. Advertisement of a path and its attributes from IS-IS protocol .................................. 26
   3.6. Advertisement of a path and its attributes from
1. Introduction

Segment Routing (SR) allows for a flexible definition of end-to-end paths by combining sub-paths called "segments". A segment can represent any instruction, topological or service-based. A segment can have a local semantic to an SR node or global within a domain. Within IGP topologies an SR path is encoded as a sequence of topological sub-paths, called "IGP segments". These segments are advertised by the link-state routing protocols (IS-IS, OSPF and OSPFv3).

Two types of IGP segments are defined, Prefix segments and Adjacency segments. Prefix segments, by default, represent an ECMP-aware shortest-path to a prefix, as per the state of the IGP topology. Adjacency segments represent a hop over a specific adjacency between two nodes in the IGP. A prefix segment is typically a multi-hop path while an adjacency segment, in most of the cases, is a one-hop path. [I-D.ietf-spring-segment-routing].

When Segment Routing is enabled in an IGP domain, segments are advertised in the form of Segment Identifiers (SIDs). The IGP link-state routing protocols have been extended to advertise SIDs and other SR-related information. IGP extensions are described in: IS-IS [I-D.ietf-isis-segment-routing-extensions], OSPFv2 [I-D.ietf-ospf-segment-routing-extensions] and OSPFv3 [I-D.ietf-ospf-ospfv3-segment-routing-extensions]. Using these extensions, Segment Routing can be enabled within an IGP domain.
Segment Routing (SR) allows advertisement of single or multi-hop paths. The flooding scope for the IGP extensions for Segment routing is IGP area-wide. Consequently, the contents of a Link State Database (LSDB) or a Traffic Engineering Database (TED) has the scope of an IGP area and therefore, by using the IGP alone it is not enough to construct segments across multiple IGP Area or AS boundaries.

In order to address the need for applications that require topological visibility across IGP areas, or even across Autonomous Systems (AS), the BGP-LS address-family/sub-address-family have been defined to allow BGP to carry Link-State information. The BGP Network Layer Reachability Information (NLRI) encoding format for BGP-LS and a new BGP Path Attribute called the BGP-LS attribute are defined in [RFC7752]. The identifying key of each Link-State object, namely a node, link, or prefix, is encoded in the NLRI and the properties of the object are encoded in the BGP-LS attribute.

Figure 1 describes a typical deployment scenario. In each IGP area, one or more nodes are configured with BGP-LS. These BGP speakers form an IBGP mesh by connecting to one or more route-reflectors. This way, all BGP speakers (specifically the route-reflectors) obtain Link-State information from all IGP areas (and from other ASes from EBGP peers). An external component connects to the route-reflector to obtain this information (perhaps moderated by...
This document describes extensions to BGP-LS to advertise the SR information. An external component (e.g., a controller) then can collect SR information in the "northbound" direction across IGP areas or ASes and construct the end-to-end path (with its associated SIDs) that need to be applied to an incoming packet to achieve the desired end-to-end forwarding.

2. BGP-LS Extensions for Segment Routing

This document defines IGP SR extensions BGP-LS TLVs and Sub-TLVs. Section 2.4 and Section 2.5 illustrates the equivalent TLVs and Sub-TLVs in IS-IS, OSPF and OSPFv3 protocols.

BGP-LS [RFC7752] defines the BGP-LS NLRI that can be a Node NLRI, a Link NLRI or a Prefix NLRI. The corresponding BGP-LS attribute is a Node Attribute, a Link Attribute or a Prefix Attribute. BGP-LS [RFC7752] defines the TLVs that map link-state information to BGP-LS NLRI and the BGP-LS attribute. This document adds additional BGP-LS attribute TLVs in order to encode SR information.

2.1. Node Attributes

The following Node Attribute TLVs are defined:

<table>
<thead>
<tr>
<th>TLV Code Point</th>
<th>Description</th>
<th>Length</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1034</td>
<td>SR Capabilities</td>
<td>variable</td>
<td>Section 2.1.1</td>
</tr>
<tr>
<td>1035</td>
<td>SR Algorithm</td>
<td>variable</td>
<td>Section 2.1.2</td>
</tr>
<tr>
<td>1036</td>
<td>SR Local Block</td>
<td>variable</td>
<td>Section 2.1.3</td>
</tr>
<tr>
<td>1037</td>
<td>SRMS Preference</td>
<td>variable</td>
<td>Section 2.1.4</td>
</tr>
</tbody>
</table>

Table 1: Node Attribute TLVs

These TLVs can ONLY be added to the Node Attribute associated with the Node NLRI that originates the corresponding SR TLV.

2.1.1. SR-Capabilities TLV

The SR Capabilities sub-TLV has following format:
Type: TBD, suggested value 1034.
Length: Variable.

Flags: 1 octet of flags as defined in
[I-D.ietf-isis-segment-routing-extensions] and
[I-D.ietf-ospf-ospfv3-segment-routing-extensions].

One or more entries, each of which have the following format:

Range Size: 3 octet value indicating the number of labels in
the range.

SID/Label sub-TLV (as defined in Section 2.3.7.2).

2.1.2. SR-Algorithm TLV

The SR-Algorithm TLV has the following format:

where:

Type: TBD, suggested value 1035.
Length: Variable.
Algorithm: 1 octet identifying the algorithm.

2.1.3. SR Local Block TLV

The SR Local Block (SRLB) Sub-TLV contains the range of labels the node has reserved for local SIDs. Local SIDs are used, e.g., in IGP (IS-IS, OSPF) for Adjacency-SIDs, and may also be allocated by other components than IGP protocols. As an example, an application or a controller may instruct a node to allocate a specific local SID. Therefore, in order for such applications or controllers to know the range of local SIDs available, it is required that the node advertises its SRLB.

The SRLB TLV has the following format:

```
  0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5   6   7   8   9   0   1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Type            |               Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Flags    |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Range Size                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                SID/Label Sub-TLV (variable)                 //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Type: TBD, suggested value 1036.

Length: Variable.

Flags: 1 octet of flags. None are defined at this stage.

One or more entries, each of which have the following format:

Range Size: 3 octet value indicating the number of labels in the range.

SID/Label sub-TLV (as defined in Section 2.3.7.2).

2.1.4. SRMS Preference TLV

The Segment Routing Mapping Server (SRMS) Preference sub-TLV is used in order to associate a preference with SRMS advertisements from a particular source.
The SRMS Preference sub-TLV has following format:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type        |    Length     | Preference    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: TBD, suggested value 1037.

Length: 1.


The use of the SRMS Preference TLV is defined in
[I-D.ietf-isis-segment-routing-extensions].

2.2.  Link Attribute TLVs

The following Link Attribute TLVs are are defined:

<table>
<thead>
<tr>
<th>TLV Code</th>
<th>Description</th>
<th>Length</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1099</td>
<td>Adjacency Segment Identifier (Adj-SID) TLV</td>
<td>variable</td>
<td>Section 2.2.1</td>
</tr>
<tr>
<td>1100</td>
<td>LAN Adjacency Segment Identifier (Adj-SID) TLV</td>
<td>variable</td>
<td>Section 2.2.2</td>
</tr>
</tbody>
</table>

Table 2: Link Attribute TLVs

These TLVs can ONLY be added to the Link Attribute associated with
the link whose local node originates the corresponding TLV.

For a LAN, normally a node only announces its adjacency to the IS-IS
pseudo-node (or the equivalent OSPF Designated and Backup Designated
Routers)[I-D.ietf-isis-segment-routing-extensions]. The LAN
Adjacency Segment TLV allows a node to announce adjacencies to all
other nodes attached to the LAN in a single instance of the BGP-LS
Link NLRI. Without this TLV, the corresponding BGP-LS link NLRI
would need to be originated for each additional adjacency in order to
advertise the SR TLVs for these neighbor adjacencies.
2.2.1. Adjacency SID TLV

The Adjacency SID (Adj-SID) TLV has the following format:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Type            |              Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Flags         |     Weight    |             Reserved          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   SID/Label/Index (variable)                  |
+---------------------------------------------------------------+
```

where:

Type: TBD, suggested value 1099.
Length: Variable.
Flags. 1 octet field of following flags as defined in
[I-D.ietf-isis-segment-routing-extensions],
[I-D.ietf-ospf-segment-routing-extensions] and
[I-D.ietf-ospf-ospfv3-segment-routing-extensions].
Weight: Weight used for load-balancing purposes.
SID/Index/Label: Label or index value depending on the flags
setting as defined in [I-D.ietf-isis-segment-routing-extensions],
[I-D.ietf-ospf-segment-routing-extensions] and
[I-D.ietf-ospf-ospfv3-segment-routing-extensions].

2.2.2. LAN Adjacency SID TLV

The LAN Adjacency SID (LAN-Adj-SID-SID) has the following format:
where:

Type: TBD, suggested value 1100.
Length: Variable.
Flags. 1 octet field of following flags as defined in
[I-D.ietf-isis-segment-routing-extensions],
[I-D.ietf-ospf-segment-routing-extensions] and
[I-D.ietf-ospf-ospfv3-segment-routing-extensions].
Weight: Weight used for load-balancing purposes.
SID/Index/Label: Label or index value depending on the flags
setting as defined in [I-D.ietf-isis-segment-routing-extensions],
[I-D.ietf-ospf-segment-routing-extensions] and
[I-D.ietf-ospf-ospfv3-segment-routing-extensions].

2.3. Prefix Attribute TLVs

The following Prefix Attribute TLVs and Sub-TLVs are defined:
### Table 3: Prefix Attribute TLVs

<table>
<thead>
<tr>
<th>TLV Code Point</th>
<th>Description</th>
<th>Length</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1158</td>
<td>Prefix SID</td>
<td>variable</td>
<td>Section 2.3.1</td>
</tr>
<tr>
<td>1159</td>
<td>Range</td>
<td>variable</td>
<td>Section 2.3.5</td>
</tr>
<tr>
<td>1160</td>
<td>Binding SID</td>
<td>variable</td>
<td>Section 2.3.6</td>
</tr>
<tr>
<td>1169</td>
<td>IPv6 Prefix SID</td>
<td>variable</td>
<td>Section 2.3.2</td>
</tr>
<tr>
<td>1170</td>
<td>IGP Prefix Attributes</td>
<td>variable</td>
<td>Section 2.3.3</td>
</tr>
<tr>
<td>1171</td>
<td>Source Router-ID</td>
<td>variable</td>
<td>Section 2.3.4</td>
</tr>
</tbody>
</table>

### Table 4: Prefix Attribute - Binding SID Sub-TLVs

#### 2.3.1. Prefix-SID TLV

The Prefix-SID TLV can ONLY be added to the Prefix Attribute whose local node in the corresponding Prefix NLRI is the node that originates the corresponding SR TLV.

The Prefix-SID has the following format:
where:

Type: TBD, suggested value 1158.

Length: Variable

Algorithm: 1 octet value identify the algorithm.

SID/Index/Label: Label or index value depending on the flags setting as defined in [I-D.ietf-isis-segment-routing-extensions], [I-D.ietf-ospf-segment-routing-extensions] and [I-D.ietf-ospf-ospfv3-segment-routing-extensions].

The Prefix-SID TLV includes a Flags field. In the context of BGP-LS, the Flags field format and the semantic of each individual flag MUST be taken from the corresponding source protocol (i.e.: the protocol of origin of the Prefix-SID being advertised in BGP-LS).

IS-IS Prefix-SID flags are defined in [I-D.ietf-isis-segment-routing-extensions] section 2.1.

OSPF Prefix-SID flags are defined in [I-D.ietf-ospf-segment-routing-extensions] section 5.

OSPFv3 Prefix-SID flags are defined in [I-D.ietf-ospf-ospfv3-segment-routing-extensions] section 5.

2.3.2. IPv6 Prefix-SID TLV

The IPv6 Prefix-SID TLV can ONLY be added to the Prefix Attribute whose local node in the corresponding Prefix NLRI is the node that originates the corresponding SR TLV.

The IPv6 Prefix-SID has the following format:
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type               |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Flags              |    Algorithm  |              //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+              //
//                             Sub-TLVs                        //
//                                                             //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
where:

Type: TBD, suggested value 1169.

Length: 3 + length of Sub-TLVs.

Flags: 2 octet field of flags. None of them is defined at this stage.

Algorithm: 1 octet value identify the algorithm as defined in [I-D.previdi-isis-ipv6-prefix-sid].

Sub-TLVs: additional information encoded into the IPv6 Prefix-SID Sub-TLV as defined in [I-D.previdi-isis-ipv6-prefix-sid].

The IPv6 Prefix-SID TLV is defined in [I-D.previdi-isis-ipv6-prefix-sid].

2.3.3.  IGP Prefix Attributes TLV

The IGP Prefix Attribute TLV carries IPv4/IPv6 prefix attribute flags as defined in [RFC7684] and [RFC7794].

The IGP Prefix Attribute TLV has the following format:

0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type               |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Flags (variable)             |              //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

where:

Type: TBD, suggested value 1170.
Length: variable.

Flags: a variable length flag field (according to the length field). Flags are routing protocol specific (OSPF and IS-IS). OSPF flags are defined in [RFC7684] and IS-IS flags are defined in [RFC7794]. The receiver of the BGP-LS update, when inspecting the IGP Prefix Attribute TLV, MUST check the Protocol-ID of the NLRI and refer to the protocol specification in order to parse the flags.

2.3.4. Source Router Identifier (Source Router-ID) TLV

The Source Router-ID TLV contains the IPv4 or IPv6 Router-ID of the originator as defined in [RFC7794]. While defined in the IS-IS protocol, the Source Router-ID TLV may be used to carry the OSPF Router-ID of the prefix originator.

The Source Router-ID TLV has the following format:

```
0                   1                   2                   3
           +--------------------------------------------------+
           |            Type               |            Length             |
           +--------------------------------------------------+
           //                  IPv4/IPv6 Address (Router-ID)              //
           +--------------------------------------------------+
```

where:

Type: TBD, suggested value 1171.

Length: 4 or 16.

IPv4/IPv6 Address: 4 octet IPv4 address or 16 octet IPv6 address.

The semantic of the Source Router-ID TLV is defined in [RFC7794].

2.3.5. Range TLV

The Range TLV can ONLY be added to the Prefix Attribute whose local node in the corresponding Prefix NLRI is the node that originates the corresponding SR TLV.

When the range TLV is used in order to advertise a path to a prefix or a range of prefix-to-SID mappings, the Prefix-NLRI the Range TLV is attached to MUST be advertised as a non-routing prefix where no IGP metric TLV (TLV 1095) is attached.
The format of the Range TLV is as follows:

```
+---------------+---------------+---------------+---------------+
|                |                |                |                |
| Type           | Length        |                |                |
| Flags          | RESERVED      | Range Size    |                |
|                |                | Sub-TLVs      |                |
+---------------+---------------+---------------+---------------+
```

Figure 2: Range TLV format

- **Type**: 1159
- **Length**: is 4.
- **Flags**: Only used when the source protocol is OSPF and defined in [I-D.ietf-ospf-segment-routing-extensions] section 4 and [I-D.ietf-ospf-ospfv3-segment-routing-extensions] section 4.

Within the Range TLV, the following SubTLVs are may be present:

- **Binding SID TLV**, defined in Section 2.3.6
- **Prefix-SID TLV**, defined in Section 2.3.1
- **SID/Label TLV**, defined in Section 2.3.7.2

**2.3.6. Binding SID TLV**

The Binding SID TLV can be used in two ways:

- o as a sub-TLV of the Range TLV
- o as a Prefix Attribute TLV

The format of the Binding SID TLV is as follows:
Figure 3: Binding SID Sub-TLV format

Type is 1160
Length is variable

Weight and Flags are mapped to Weight and Flags defined in
[I-D.ietf-isis-segment-routing-extensions] section 2.4,
[I-D.ietf-ospf-segment-routing-extensions] section 4 and

Sub-TLVs are defined in the following sections.

2.3.7. Binding SID SubTLVs

This section defines the Binding SID Sub-TLVs in BGP-LS to encode the
equivalent Sub-TLVs defined in
[I-D.ietf-isis-segment-routing-extensions],
[I-D.ietf-ospf-segment-routing-extensions] and
[I-D.ietf-ospf-ospfv3-segment-routing-extensions].

All ERO (Explicit Route Object) Sub-TLVs must immediately follow the
(SID)/Label Sub-TLV.

All Backup ERO Sub-TLVs must immediately follow the last ERO Sub-TLV.

2.3.7.1. Binding SID Prefix-SID Sub-TLV

When encoding IS-IS Mapping Server entries as defined in
[I-D.ietf-isis-segment-routing-extensions] the Prefix-SID TLV defined
in Section 2.3.1 is used as Sub-TLV in the Binding TLV.
2.3.7.2. SID/Label Sub-TLV

The SID/Label TLV has following format:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Type            |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      SID/Label (variable)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: TBD, suggested value 1161.
Length: Variable, 3 or 4 bytes
SID/Label: If length is set to 3, then the 20 rightmost bits represent a label. If length is set to 4, then the value represents a 32 bit SID.

The receiving router MUST ignore the SID/Label Sub-TLV if the length is other then 3 or 4.

2.3.7.3. ERO Metric Sub-TLV

The ERO Metric Sub-TLV has following format:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Metric (4 octets)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

ERO Metric Sub-TLV format

where:

Type: TBD, suggested value 1162.
Length: Always 4
Metric: A 4 octet metric representing the aggregate IGP or TE path cost.
2.3.7.4. IPv4 ERO Sub-TLV

The ERO Sub-TLV has following format:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Type            |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Flags    |                   Reserved                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        IPv4 Address (4 octets)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv4 ERO Sub-TLV format

where:

- **Type**: TBD, suggested value 1163
- **Length**: 8 octets
- **Flags**: 1 octet of flags as defined in:
  - [I-D.ietf-isis-segment-routing-extensions],
  - [I-D.ietf-ospf-segment-routing-extensions] and
  - [I-D.ietf-ospf-ospfv3-segment-routing-extensions].

**IPv4 Address** - the address of the explicit route hop.

2.3.7.5. IPv6 ERO Sub-TLV

The IPv6 ERO Sub-TLV has following format:
IPv6 ERO Sub-TLV format

where:

Type: TBD, suggested value 1164
Length: 20 octets
Flags: 1 octet of flags as defined in:
[I-D.ietf-isis-segment-routing-extensions],
[I-D.ietf-ospf-segment-routing-extensions] and
[I-D.ietf-ospf-ospfv3-segment-routing-extensions].
IPv6 Address - the address of the explicit route hop.

2.3.7.6. Unnumbered Interface ID ERO Sub-TLV

The Unnumbered Interface-ID ERO Sub-TLV has following format:
where:

Unnumbered Interface ID ERO Sub-TLV format

Type: TBD, suggested value 1165.


Flags: 1 octet of flags as defined in:
[I-D.ietf-isis-segment-routing-extensions],
[I-D.ietf-ospf-segment-routing-extensions] and
[I-D.ietf-ospf-ospfv3-segment-routing-extensions].

Router-ID: Router-ID of the next-hop.

Interface ID: is the identifier assigned to the link by the router
specified by the Router-ID.

2.3.7.7. IPv4 Backup ERO Sub-TLV

The IPv4 Backup ERO Sub-TLV has following format:

where:
Type: TBD, suggested value 1166.

Length: 8 octets

Flags: 1 octet of flags as defined in:
[I-D.ietf-isis-segment-routing-extensions],
[I-D.ietf-ospf-segment-routing-extensions] and
[I-D.ietf-ospf-ospfv3-segment-routing-extensions].

IPv4 Address: Address of the explicit route hop.

2.3.7.8. IPv6 Backup ERO Sub-TLV

The IPv6 Backup ERO Sub-TLV has following format:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Flags      |                   Reserved                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
IPv6 Backup ERO Sub-TLV format

where:

Type: TBD, suggested value 1167.

Length: 8 octets

Flags: 1 octet of flags as defined in:
[I-D.ietf-isis-segment-routing-extensions],
[I-D.ietf-ospf-segment-routing-extensions] and
[I-D.ietf-ospf-ospfv3-segment-routing-extensions].

IPv6 Address: Address of the explicit route hop.
2.3.7.9. Unnumbered Interface ID Backup ERO Sub-TLV

The Unnumbered Interface ID Backup ERO Sub-TLV has following format:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Type                 |             Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Flags            |                   Reserved   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                     Router ID (IPv4, 4 octet or IPv6, 16 octets)  //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Interface ID                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Unnumbered Interface ID Backup ERO Sub-TLV format

where:

- **Type**: TBD, suggested value 1168.
- **Length**: Variable (12 for IPv4 Router-ID or 24 for IPv6 Router-ID).
- **Flags**: 1 octet of flags as defined in: [I-D.ietf-isis-segment-routing-extensions], [I-D.ietf-ospf-segment-routing-extensions] and [I-D.ietf-ospf-ospfv3-segment-routing-extensions].
- **Router-ID**: Router-ID of the next-hop.
- **Interface ID**: Identifier assigned to the link by the router specified by the Router-ID.

2.4. Equivalent IS-IS Segment Routing TLVs/Sub-TLVs

This section illustrate the IS-IS Segment Routing Extensions TLVs and Sub-TLVs mapped to the ones defined in this document.

The following table, illustrates for each BGP-LS TLV, its equivalence in IS-IS.
<table>
<thead>
<tr>
<th>TLV Code</th>
<th>Description</th>
<th>Length</th>
<th>IS-IS TLV/Sub-TLV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1034</td>
<td>SR Capabilities</td>
<td>variable</td>
<td>2 [1]</td>
</tr>
<tr>
<td>1035</td>
<td>SR Algorithm</td>
<td>variable</td>
<td>19 [2]</td>
</tr>
<tr>
<td>1099</td>
<td>Adjacency Segment Identifier (Adj-SID) TLV</td>
<td>variable</td>
<td>31 [3]</td>
</tr>
<tr>
<td>1100</td>
<td>LAN Adjacency Segment Identifier</td>
<td>variable</td>
<td>32 [4]</td>
</tr>
<tr>
<td>1158</td>
<td>Prefix SID</td>
<td>variable</td>
<td>3 [5]</td>
</tr>
<tr>
<td>1160</td>
<td>Binding SID</td>
<td>variable</td>
<td>149 [6]</td>
</tr>
<tr>
<td>1161</td>
<td>SID/Label TLV</td>
<td>variable</td>
<td>1 [7]</td>
</tr>
<tr>
<td>1162</td>
<td>ERO Metric TLV</td>
<td>4 octets</td>
<td>10 [8]</td>
</tr>
<tr>
<td>1163</td>
<td>IPv4 ERO TLV</td>
<td>5 octets</td>
<td>11 [9]</td>
</tr>
<tr>
<td>1164</td>
<td>IPv6 ERO TLV</td>
<td>17</td>
<td>12 [10]</td>
</tr>
<tr>
<td>1165</td>
<td>Unnumbered Interface ID ERO TLV</td>
<td>variable</td>
<td>13 [11]</td>
</tr>
<tr>
<td>1166</td>
<td>IPv4 Backup ERO TLV</td>
<td>5 octets</td>
<td>14 [12]</td>
</tr>
<tr>
<td>1167</td>
<td>IPv6 Backup ERO TLV</td>
<td>17</td>
<td>15 [13]</td>
</tr>
<tr>
<td>1168</td>
<td>Unnumbered Interface ID Backup ERO TLV</td>
<td>variable</td>
<td>16 [14]</td>
</tr>
<tr>
<td>1169</td>
<td>IPv6 Prefix SID</td>
<td>variable</td>
<td>5 [15]</td>
</tr>
<tr>
<td>1170</td>
<td>IGP Prefix Attributes</td>
<td>variable</td>
<td>4 [16]</td>
</tr>
<tr>
<td>1171</td>
<td>Source Router ID</td>
<td>variable</td>
<td>11/12 [17]</td>
</tr>
</tbody>
</table>

Table 5: IS-IS Segment Routing Extensions TLVs/Sub-TLVs

2.5. Equivalent OSPF/OSPFv3 Segment Routing TLVs/Sub-TLVs

This section illustrates the OSPF and OSPFv3 Segment Routing Extensions TLVs and Sub-TLVs mapped to the ones defined in this document.

The following table illustrates for each BGP-LS TLV, its equivalence in OSPF and OSPFv3.
<table>
<thead>
<tr>
<th>TLV Code Point</th>
<th>Description</th>
<th>Length</th>
<th>OSPF TLV / Sub-TLV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1034</td>
<td>SR Capabilities</td>
<td>variable</td>
<td>9 [18]</td>
</tr>
<tr>
<td>1035</td>
<td>SR Algorithm</td>
<td>variable</td>
<td>8 [19]</td>
</tr>
<tr>
<td>1099</td>
<td>Adjacency Segment</td>
<td>variable</td>
<td>2 [20]</td>
</tr>
<tr>
<td></td>
<td>Identifier (Adj-SID) TLV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1100</td>
<td>LAN Adjacency Segment</td>
<td>variable</td>
<td>3 [21]</td>
</tr>
<tr>
<td></td>
<td>Identifier (Adj-SID) TLV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1158</td>
<td>Prefix SID</td>
<td>variable</td>
<td>2 [22]</td>
</tr>
<tr>
<td>1161</td>
<td>SID/Label TLV</td>
<td>variable</td>
<td>1 [23]</td>
</tr>
<tr>
<td>1162</td>
<td>ERO Metric TLV</td>
<td>4 octets</td>
<td>8 [24]</td>
</tr>
<tr>
<td>1163</td>
<td>IPv4 ERO TLV</td>
<td>8 octets</td>
<td>4 [25]</td>
</tr>
<tr>
<td>1165</td>
<td>Unnumbered Interface ID ERO TLV</td>
<td>12 octets</td>
<td>5 [26]</td>
</tr>
<tr>
<td>1166</td>
<td>IPv4 Backup ERO TLV</td>
<td>8 octets</td>
<td>6 [27]</td>
</tr>
<tr>
<td>1167</td>
<td>Unnumbered Interface ID Backup ERO TLV</td>
<td>octets</td>
<td>7 [28]</td>
</tr>
<tr>
<td>1167</td>
<td>Unnumbered Interface ID Backup ERO TLV</td>
<td>octets</td>
<td>7 [29]</td>
</tr>
</tbody>
</table>

Table 6: OSPF Segment Routing Extensions TLVs/Sub-TLVs
3. Procedures

The following sections describe the different operations for the propagation of SR TLVs into BGP-LS.

3.1. Advertisement of a IS-IS Prefix SID TLV

The advertisement of a IS-IS Prefix SID TLV has following rules:

The IS-IS Prefix-SID is encoded in the BGP-LS Prefix Attribute Prefix-SID as defined in Section 2.3.1. The flags in the Prefix-SID TLV have the semantic defined in [I-D.ietf-isis-segment-routing-extensions] section 2.1.

3.2. Advertisement of a OSPF/OSPFv3 Prefix-SID TLV

The advertisement of a OSPF/OSPFv3 Prefix-SID TLV has following rules:

The OSPF (or OSPFv3) Prefix-SID is encoded in the BGP-LS Prefix Attribute Prefix-SID as defined in Section 2.3.1. The flags in
the Prefix-SID TLV have the semantic defined in
[I-D.ietf-ospf-segment-routing-extensions] section 5 or

3.3. Advertisement of a range of prefix-to-SID mappings in OSPF

The advertisement of a range of prefix-to-SID mappings in OSPF has
following rules:

The OSPF/OSPFv3 Extended Prefix Range TLV is encoded in the BGP-LS
Prefix Attribute Range TLV as defined in Section 2.3.5. The flags
of the Range TLV have the semantic mapped to the definition in
[I-D.ietf-ospf-segment-routing-extensions] section 4 or
[I-D.ietf-ospf-ospfv3-segment-routing-extensions] section 4. The
Prefix-SID from the original OSPF Prefix SID Sub-TLV is encoded
using the BGP-LS Prefix Attribute Prefix-SID as defined in
Section 2.3.1 with the flags set according to the definition in
[I-D.ietf-ospf-segment-routing-extensions] section 5 or

3.4. Advertisement of a range of IS-IS SR bindings

The advertisement of a range of IS-IS SR bindings has following
rules:

In IS-IS the Mapping Server binding ranges are advertised using
the Binding TLV. The IS-IS Binding TLV is encoded in the BGP-LS
Prefix Attribute Range TLV as defined in Section 2.3.5 using the
Binding Sub-TLV as defined in Section 2.3.6. The flags in the
Range TLV are all set to zero on transmit and ignored on
reception. The range value from the original IS-IS Binding TLV is
encoded in the Range TLV "Range" field.

3.5. Advertisement of a path and its attributes from IS-IS protocol

The advertisement of a Path and its attributes is described in
[I-D.ietf-isis-segment-routing-extensions] section 2.4 and has
following rules:

The original Binding SID TLV (from IS-IS) is encoded into the BGP-
LS Range TLV defined in Section 2.3.5 using the Binding Sub-TLV as
defined in Section 2.3.6. The set of Sub-TLVs from the original
IS-IS Binding TLV are encoded as Sub-TLVs of the BGP-LS Binding
TLV as defined in Section 2.3.6. This includes the SID/Label TLV
defined in Section 2.3.
3.6. Advertisement of a path and its attributes from OSPFv2/OSPFv3 protocol

The advertisement of a Path and its attributes is described in [I-D.ietf-ospf-segment-routing-extensions] section 6 and [I-D.ietf-ospf-ospfv3-segment-routing-extensions] section 6 and has following rules:

Advertisement of a path for a single prefix: the original Binding SID TLV (from OSPFv2/OSPFv3) is encoded into the BGP-LS Prefix Attribute Binding TLV as defined in Section 2.3.6. The set of Sub-TLVs from the original OSPFv2/OSPFv3 Binding TLV are encoded as Sub-TLVs of the BGP-LS Binding TLV as defined in Section 2.3.6. This includes the SID/Label TLV defined in Section 2.3.

Advertisement of an SR path for range of prefixes: the OSPF/OSPFv3 Extended Prefix Range TLV is encoded in the BGP-LS Prefix Attribute Range TLV as defined in Section 2.3.5. The original OSPFv2/OSPFv3 Binding SID TLV is encoded into the BGP-LS Binding Sub-TLV as defined in Section 2.3.6. The set of Sub-TLVs from the original OSPFv2/OSPFv3 Binding TLV are encoded as Sub-TLVs of the BGP-LS Binding TLV as defined in Section 2.3.6. This includes the SID/Label TLV defined in Section 2.3.

4. IANA Considerations

This document requests assigning code-points from the registry for BGP-LS attribute TLVs based on table Table 8.

4.1. TLV/Sub-TLV Code Points Summary

This section contains the global table of all TLVs/Sub-TLVs defined in this document.
<table>
<thead>
<tr>
<th>TLV Code Point</th>
<th>Description</th>
<th>Length</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1034</td>
<td>SR Capabilities</td>
<td>variable</td>
<td>Section 2.1.1</td>
</tr>
<tr>
<td>1035</td>
<td>SR Algorithm</td>
<td>variable</td>
<td>Section 2.1.2</td>
</tr>
<tr>
<td>1036</td>
<td>SR Local Block</td>
<td>variable</td>
<td>Section 2.1.3</td>
</tr>
<tr>
<td>1037</td>
<td>SRMS Preference</td>
<td>variable</td>
<td>Section 2.1.4</td>
</tr>
<tr>
<td>1099</td>
<td>Adjacency Segment</td>
<td>variable</td>
<td>Section 2.2.1</td>
</tr>
<tr>
<td></td>
<td>Identifier (Adj-SID) TLV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1100</td>
<td>LAN Adjacency Segment</td>
<td>variable</td>
<td>Section 2.2.2</td>
</tr>
<tr>
<td></td>
<td>Identifier (Adj-SID) TLV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1158</td>
<td>Prefix SID</td>
<td>variable</td>
<td>Section 2.3.1</td>
</tr>
<tr>
<td>1159</td>
<td>Range</td>
<td>variable</td>
<td>Section 2.3.5</td>
</tr>
<tr>
<td>1160</td>
<td>Binding SID</td>
<td>variable</td>
<td>Section 2.3.6</td>
</tr>
<tr>
<td>1161</td>
<td>SID/Label TLV</td>
<td>variable</td>
<td>Section 2.3.7.2</td>
</tr>
<tr>
<td>1162</td>
<td>ERO Metric TLV</td>
<td>4 octets</td>
<td>1 [43]</td>
</tr>
<tr>
<td>1163</td>
<td>IPv4 ERO TLV</td>
<td>8 octets</td>
<td>1 [44]</td>
</tr>
<tr>
<td>1164</td>
<td>IPv6 ERO TLV</td>
<td>20 octets</td>
<td>1 [45]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1165</td>
<td>Unnumbered Interface ID</td>
<td>12 octets</td>
<td>1 [46]</td>
</tr>
<tr>
<td></td>
<td>ERO TLV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1166</td>
<td>IPv4 Backup ERO TLV</td>
<td>8 octets</td>
<td>1 [47]</td>
</tr>
<tr>
<td>1167</td>
<td>IPv6 Backup ERO TLV</td>
<td>20 octets</td>
<td>1 [48]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1168</td>
<td>Unnumbered Interface ID</td>
<td>12 octets</td>
<td>1 [49]</td>
</tr>
<tr>
<td></td>
<td>Backup ERO TLV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1169</td>
<td>IPv6 Prefix SID</td>
<td>variable</td>
<td>Section 2.3.2</td>
</tr>
<tr>
<td>1170</td>
<td>IGP Prefix Attributes</td>
<td>variable</td>
<td>Section 2.3.3</td>
</tr>
<tr>
<td>1171</td>
<td>Source Router-ID</td>
<td>variable</td>
<td>Section 2.3.4</td>
</tr>
</tbody>
</table>

Table 8: Summary Table of TLV/Sub-TLV Codepoints

5. Manageability Considerations

This section is structured as recommended in [RFC5706].

5.1. Operational Considerations

5.1.1. Operations

Existing BGP and BGP-LS operational procedures apply. No additional operation procedures are defined in this document.
6. Security Considerations

Procedures and protocol extensions defined in this document do not affect the BGP security model. See the ‘Security Considerations’ section of [RFC4271] for a discussion of BGP security. Also refer to [RFC4272] and [RFC6952] for analysis of security issues for BGP.

7. Contributors

The following people have substantially contributed to the editing of this document:

Acee Lindem  
Cisco Systems  
Email: acee@cisco.com

Saikat Ray  
Individual  
Email: raysaikat@gmail.com

8. Acknowledgements

The authors would like to thank Les Ginsberg for the review of this document.

9. References

9.1. Normative References

[I-D.ietf-isis-segment-routing-extensions]  

[I-D.ietf-ospf-ospfv3-segment-routing-extensions]  

[I-D.ietf-ospf-segment-routing-extensions]  
9.2. Informative References


9.3. URIs


Authors’ Addresses

Stefano Previdi (editor)
Cisco Systems, Inc.
Via Del Serafico, 200
Rome  00142
Italy

Email: sprevidi@cisco.com

Peter Psenak
Cisco Systems, Inc.
Apollo Business Center
Mlynske nivy 43
Bratislava  821 09
Slovakia

Email: ppsenak@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Brussels
Belgium

Email: cfilsfil@cisco.com

Hannes Gredler
RtBrick Inc.

Email: hannes@rtbrick.com
Abstract

BGP flow specification version 1 (FSv1), defined in RFC 8955, RFC 8956, and RFC 9117 describes the distribution of traffic filter policy (traffic filters and actions) distributed via BGP. Multiple applications have used BGP FSv1 to distribute traffic filter policy. These applications include the following: mitigation of denial of service (DoS), enabling traffic filtering in BGP/MPLS VPNs, centralized traffic control of router firewall functions, and SFC traffic insertion.

During the deployment of BGP FSv1 a number of issues were detected due to lack of consistent TLV encoding for rules for flow specifications, lack of user ordering of filter rules and/or actions, and lack of clear definition of interaction with BGP peers not supporting FSv1. Version 2 of the BGP flow specification (FSv2) protocol addresses these features. In order to provide a clear demarcation between FSv1 and FSv2, a different NLRI encapsulates FSv2.

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 8 August 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.

Table of Contents

1. Introduction .................................. 4
   1.1. Definitions and Acronyms ...................... 5
   1.2. RFC 2119 language ........................... 6
2. Flow Specification ............................. 6
   2.1. Flow Specification v1 (FSv1) Overview ......... 7
   2.2. Flow Specification v2 (FSv2) Overview .......... 9
3. FSv2 Filters and Actions ........................ 12
   3.1. IP header SubTLV (type=1) ..................... 14
      3.1.1. IP Destination Prefix (type = 1) .......... 16
      3.1.2. IP Source Prefix (type = 2) ............... 16
      3.1.3. IP Protocol (type = 3) ..................... 17
      3.1.4. Port (type = 4) .......................... 17
      3.1.5. Destination Port (type = 5) ............... 18
      3.1.6. Source Port (type = 6) .................... 18
      3.1.7. ICMP Type (type = 7) ....................... 18
      3.1.8. ICMP Code (type = 8) ....................... 19
      3.1.9. TCP Flags (type = 9) ....................... 19
      3.1.10. Packet length (type = 10 (0x0A)) .......... 19
      3.1.11. DSCP (Differentiated Services Code Point)(type = 11 (0x0B)) .......................... 20
      3.1.12. Fragment (type = 12 (0x0C)) ............... 20
      3.1.13. Flow Label(type = 13 (0x0D)) .............. 21
      3.1.14. TTL (type=14 (0x0E)) ..................... 21
      3.1.15. Parts of SID (type = 15 (0xF)) .......... 21
      3.1.16. MPLS Label Match1 (type=16, 0x10) ........ 24
      3.1.17. MPLS Label Match 2: Experimental bits match on top label (Type=17 (0x11)) ............. 25
   3.2. Encoding of FSv2 Actions (type=2) ............ 26
      3.2.1. Action Chain operation (ACO) (1, 0x01) ...... 28
      3.2.2. Traffic Actions per interface set (TAIS) (2, 0x02) .... 29
3.2.3. Traffic rate limited by bytes (TRB) (6, 0x06) . . . . 30
3.2.4. Traffic Action (TA) (7, 0x07) . . . . . . . . . . . . 30
3.2.5. Redirect to IPv4 (RDIPv4) (8, 0x08) . . . . . . . . . . 31
3.2.6. Traffic marking (TM) (9, 0x09) . . . . . . . . . . . . 32
3.2.7. Traffic rate limited by packets (TRP) (12, 0xC) . . . 33
3.2.8. Traffic redirect to IPv6 (RDIPv6) (13, 0xD) . . . . . 33
3.2.9. Traffic insertion in SFC (TISFC) (14, 0xE) . . . . . 34
3.2.10. Flow Specification Redirect to Indirection-ID (RDIID) (15, 0x0F) . . . . . . . . . . . . . . . . . . . . . 35
3.2.11. MPLS Label Action (MPLSLA) (16, 0x10) . . . . . . 36
3.2.12. VLAN action (VLAN) (22, 0x16) . . . . . . . . . . . . 37
3.2.13. TPID action (TPID) (23, 0x17) . . . . . . . . . . . . 39
3.3. Extended Community vs. Action SubTLV formats . . . . . 39
3.4. L2 Traffic Rules . . . . . . . . . . . . . . . . . . . . . . 42
3.5. SFC Traffic Rules . . . . . . . . . . . . . . . . . . . . . 43
3.6. BGP/MPLS VPN IP Traffic Rules . . . . . . . . . . . . . . 45
3.7. BGP/MPLS VPN L2 Traffic Rules . . . . . . . . . . . . . . 45
3.8. Encoding of Actions passed in Wide Communities . . . . 46
4. Validation of FSv2 NLRI . . . . . . . . . . . . . . . . . . . 47
4.1. Validation of FS NLRI (FSv1 or FSv2) . . . . . . . . . . 47
4.2. Validation of Flow Specification Actions . . . . . . . . . 49
4.3. Error handling and Validation . . . . . . . . . . . . . . . 50
5. Ordering for Flow Specification v2 (FSv2) . . . . . . . . . . 50
5.1. Ordering of FSv2 NLRI Filters . . . . . . . . . . . . . . 50
5.2. Ordering of the Actions . . . . . . . . . . . . . . . . . . 52
5.2.1. Action Chain Operation (ACO) . . . . . . . . . . . . 52
5.2.2. Summary of FSv2 ordering . . . . . . . . . . . . . . . 55
6. Ordering of FS filters for BGP Peers support FSv1 and FSv2 . 56
7. Scalability and Aspirations for FSv2 . . . . . . . . . . . . . 58
8. Optional Security Additions . . . . . . . . . . . . . . . . . . 59
8.1. BGP FSv2 and BGPSEC . . . . . . . . . . . . . . . . . . 59
8.2. BGP FSv2 with ROA . . . . . . . . . . . . . . . . . . . . 60
9. IANA Considerations . . . . . . . . . . . . . . . . . . . . . . 60
9.1. Flow Specification V2 SAFIs . . . . . . . . . . . . . . . . 60
9.2. BGP Capability Code . . . . . . . . . . . . . . . . . . . 61
9.3. Filter IP Component types . . . . . . . . . . . . . . . . . 61
9.4. FSv2 NLRI TLV Types . . . . . . . . . . . . . . . . . . . 62
9.5. Wide Community Assignments . . . . . . . . . . . . . . . 63
10. Security Considerations . . . . . . . . . . . . . . . . . . . 64
11. References . . . . . . . . . . . . . . . . . . . . . . . . . . 64
11.1. Normative References . . . . . . . . . . . . . . . . . . . 64
11.2. Informative References . . . . . . . . . . . . . . . . . . 67
Authors’ Addresses . . . . . . . . . . . . . . . . . . . . . . . . 68
1. Introduction

Modern IP routers have the capability to forward traffic and to classify, shape, rate limit, filter, or redirect packets based on administratively defined policies. These traffic policy mechanisms allow the operator to define match rules that operate on multiple fields within header of an IP data packet. The traffic policy allows actions to be taken upon a match to be associated with each match rule. These rules can be more widely defined as "event-condition-action" (ECA) rules where the event is always the reception of a packet.

BGP ([RFC4271]) flow specification as defined by [RFC8955], [RFC8956], [RFC9117] specifies the distribution of traffic filter policy (traffic filters and actions) via BGP to a mesh of BGP peers (IBGP and EBGP peers). The traffic filter policy is applied when packets are received on a router with the flow specification function turned on. The flow specification protocol defined in [RFC8955], [RFC8956], and [RFC9117] will be called BGP flow specification version 1 (BGP FSv1) in this draft.

Some modern IP routers also include the abilities of firewalls which can match on a sequence of packet events based on administrative policy. These firewall capabilities allow for user ordering of match rules and user ordering of actions per match.

Multiple deployed applications currently use BGP FSv1 to distribute traffic filter policy. These applications include: 1) mitigation of Denial of Service (DoS), 2) traffic filtering in BGP/MPLS VPNS, and 3) centralized traffic control for networks utilizing SDN control of router firewall functions, 4) classifiers for insertion in an SFC, and 5) filters for SRv6 (segment routing v6).

During the deployment of BGP flow specification v1, the following issues were detected:

* lack of consistent TLV encoding prevented extension of encodings,

* inability to allow user defined order for filtering rules,

* inability to order actions to provide deterministic interactions or to allow users to define order for actions, and

* no clearly defined mechanisms for BGP peers which do not support flow specification v1.
Networks currently cope with some of these issues by limiting the type of traffic filter policy sent in BGP. Current Networks do not have a good workaround/solution for applications that receive but do not understand FSv1 policies.

This document defines version 2 of the BGP flow specification protocol to address these shortcomings in BGP FSv1. Version 2 of BGP flow specification will be denoted as BGP FSv2.

BGP FSv1 as defined in [RFC8955], [RFC8956], and [RFC9117] specified 2 SAFIs (133, 134) to be used with IPv4 AFI (AFI = 1) and IPv6 AFI (AFI=2).

This document specifies 2 new SAFIs (TBD1, TBD2) for FSv2 to be used with 5 AFIs (1, 2, 6, 25, and 31) to allow user-ordered lists of traffic match filters for user-ordered traffic match actions encoded in Communities (Wide or Extended).

FSv1 and FSv2 use different AFI/SAFIs to send flow specification filters. Since BGP route selection is performed per AFI/SAFI, this approach can be termed "ships in the night" based on AFI/SAFI.

FSv1 is a critical component of deployed applications. Therefore, this specification defines how FSv2 will interact with BGP peers that support either FSv2, FSv1, FSv2 and FSv1, or neither of them. It is expected that a transition to FSv2 will occur over time as new applications require FSv2 extensibility and user-defined ordering for rules and actions or network operators tire of the restrictions of FSv1 such as error handling issues and restricted topologies.

Section 2 contains the definition of Flow specification, a short review of FSv1 and an overview of FSv2. Section 3 contains the encoding rules for FSv2 and user-based encoding sent via BGP. Section 4 describes how to validate FSv2 NLRI. Section 5 discusses how to order FSv2 rules. Section 6 covers combining FSv2 user-ordered match rules and FSv1 rules. Section 6 also discusses how to combine user-ordered actions, FSv1 actions, and default actions. Sections 7-10 address an alternate security mechanism, considerations for IANA, security in deployments, and scalability aspirations.

1.1. Definitions and Acronyms

API - Address Family Identifier

AS - Autonomous System

BGPSEC - secure BGP [RFC8205] updated by [RFC8206]
BGP Session ephemeral state - state which does not survive the loss of BGP peer session.

Configuration state - state which persist across a reboot of software module within a routing system or a reboot of a hardware routing device.

DDOs - Distributed Denial of Service.

Ephemeral state - state which does not survive the reboot of a software module, or a hardware reboot. Ephemeral state can be ephemeral configuration state or operational state.

FSv1 - Flow Specification version 1 [RFC8955] [RFC8956]

FSv2 - Flow Specification version 2 (this document)

NETCONF - The Network Configuration Protocol [RFC6241].

RESTCONF - The RESTCONF configuration Protocol [RFC8040]

RIB - Routing Information Base.

ROA - Route Origin Authentication [RFC6482]

RR - Route Reflector.

SAFI - Subsequent Address Family Identifier

1.2. RFC 2119 language

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

2. Flow Specification

A BGP Flow Specification is an n-tuple containing one or more match criteria that can be applied to IP traffic, traffic encapsulated in IP traffic or traffic associated with IP traffic. The following are examples of such traffic: IP packet or an IP packet inside a L2 packet (Ethernet), an MPLS packet, and SFC flow.

A given Flow Specification NLRI may be associated with a set of path attributes depending on the particular application, and attributes within that set may or may not include reachability information.
(e.g., NEXT_HOP). Extended Community or Wide Community attributes (well-known or AS-specific) MAY be used to encode a set of pre-determined actions.

A particular application is identified by a specific AFI/SAFI (Address Family Identifier/Subsequent Address Family Identifier) and corresponds to a distinct set of RIBs. Those RIBs should be treated independently of each other in order to assure noninterference between distinct applications.

BGP processing treats the NLRI as a key to entries in AFI/SAFI BGP databases. Entries that are placed in the Loc-RIB are then associated with a given set of semantics which are application dependent. Standard BGP mechanisms such as update filtering by NLRI or by attributes such as AS_PATH or large communities apply to the BGP Flow Specification defined NLRI-types.

Network operators can control the propagation of BGP routes by enabling or disabling the exchange of routes for a particular AFI/SAFI pair on a particular peering session. As such, the Flow Specification may be distributed to only a portion of the BGP infrastructure.

2.1. Flow Specification v1 (FSv1) Overview

The FSv1 NLRI defined in [RFC8955] and [RFC8956] include 13 match conditions encoded for the following AFI/SAFIs:

* IPv4 traffic: AFI:1, SAFI:133
* IPv6 Traffic: AFI:2, SAFI:133
* BGP/MPLS IPv4 VPN: AFI:1, SAFI: 134
* BGP/MPLS IPv6 VPN: AFI:2, SAFI: 134

If one considers the reception of the packet as an event, then BGP FSv1 describes a set of Event-MatchCondition-Action (ECA) policies where:

* event is the reception of a packet,
* condition stands for "match conditions" defined in the BGP NLRI as an n-tuple of component filters, and
* the action is either: the default condition (accept traffic), or a set of actions (1 or more) defined in Extended BGP Community values [RFC4360].
The flow specification conditions and actions combine to make up FSv1 specification rules. Each FSv1 NLRI must have a type 1 component (destination prefix). Extended Communities with FSv1 actions can be attached to a single NLRI or multiple NLRIs in a BGP message.

Within an AFI/SAFI pair, FSv1 rules are ordered based on the components in the packet (types 1-13) ordered from left-most to right-most and within the component types by value of the component. Rules are inserted in the rule list by component-based order where an FSv1 rule with existing component type has higher precedence than one missing a specific component type.

Since FSv1 specifications ([RFC8955], [RFC8956], and [RFC9117]) specify that the FSv1 NLRI MUST have a destination prefix (as component type 1) embedded in the flow specification, the FSv1 rules with destination components are ordered by IP Prefix comparison rules for IPv4 ([RFC8955]) and IPv6 ([RFC8956]). [RFC8955] specifies that more specific prefixes (aka longest match) have higher precedence than that of less specific prefixes and that for prefixes of the same length the lower IP number is selected (lowest IP value). [RFC8955] specifies that if the offsets within component 1 are the same, then the longest match and lowest IP comparison rules from [RFC8955] apply. If the offsets are different, then the lower offset has precedence.

These rules provide a set of FSv1 rules ordered by IP Destination Prefix by longest match and lowest IP address. [RFC8955] also states that the requirement for a destination prefix component "MAY be relaxed by explicit configuration" Since the rule insertions are based on comparing component types between two rules in order, this means the rules without destination prefixes are inserted after all rules which contain destination prefix component.

The actions specified in FSv1 are:

* accept packet (default),
* traffic flow limitation by bytes (0x6),
* traffic-action (0x7),
* redirect traffic (0x8),
* mark traffic (0x9), and
* traffic flow limitation by packets (12, 0xC)
Figure 1 shows a diagram of the FSv1 logical data structures with 5 rules. If FSv1 rules have destination prefix components (type=1) and FSv1 rule 5 does not have a destination prefix, then FSv1 rule 5 will be inserted in the policy after rules 1-4.

**Figure 2-1: BGP Flow Specification v1 Policy**

2.2. Flow Specification v2 (FSv2) Overview

Flow Specification v2 allows the user to order the flow specification rules and the actions associated with a rule. Each FSv2 rule may have one or more match conditions and one or more associated actions.

This FSv2 specification supports the components and actions for the following:
* IPv4 (AFI=1, SAFI=TBD1),
* IPv6 (AFI=2, SAFI=TBD2),
* L2 (AFI=6, SAFI=TDB1),
* BGP/MPLS IPv4 VPN: (AFI=1, SAFI=TBD2),
* BGP/MPLS IPv6 VPN: (AFI=2, SAFI=TBD2),
* BGP/MPLS L2VPN (AFI=25, SAFI=TDB2),
* SFC: (AFI=31, SAFI=TBD1), and
* SFC VPN (AFI=31, SAFI=TBD2).

The FSv2 specification for tunnel traffic is outside the scope of this specification. The FSv1 specification for tunneled traffic is in [I-D.ietf-idr-flowspec-nvo3].

FSv2 operates in the ships-in-the night model with FSv1 so network operators can manipulate which the distribution of FSv2 and FSv1 using configuration parameters. Since the lack of deterministic ordering was an FSv1 problem, this specification provides rules and protocol features to keep filters in a deterministic order between FSv1 and FSv2.

The basic principles regarding ordering of flow specification filter rules are:

1) Rule-0 (zero) is defined to be 0/0 with the "permit-all" action.

2) FSv2 rules are ordered based on user-specified order.
   - The user-specified order is carried in the FSv2 NLRI and a numerical lower value takes precedence over a numerically higher value. For rules received with the same order value, the FSv1 rules apply (order by component type and then by value of the components).

3) FSv2 rules are added starting with Rule 1 and FSv1 rules are added after FSv2 rules
   - For example, BGP Peer A has FSv2 data base with 10 FSv2 rules (1-10). FSv1 user number is configured to start at 301 so 10 FSv1 rules are added at 301-310.
4) An FSv2 peer may receive BGP NLRI routes from a FSv1 peer or a BGP peer that does not support FSv1 or FSv2. The capabilities sent by a BGP peer indicate whether the AFI/SAFI can be received (FSv1 NLRI or FSv2 NLRI).

5) Associate a chain of actions to rules based on user-defined action number (1-n). (optional)

- If no actions are associated with a filter rule, the default is to drop traffic the filter rules match

- An action chain of 1-n actions can be associated with a set of filter rules can via Extended Communities or Wide Communities. Only Wide Communities can associate a user-defined order for the actions. Extended Community actions occur after actions with a user specified order (see section 5.2 for details).

Figure 2-2 provides a logical diagram of the FSv2 structure
3. FSv2 Filters and Actions

The BGP FSv2 uses an NRLI with the format for AFIs for IPv4 (AFI = 1), IPv6 (AFI = 2), L2 (AFI = 6), L2VPN (AFI=25), and SFC (AFI=31) with two following SAFIs to support transmission of the flow specification which supports user ordering of traffic filters and actions for IP traffic and IP VPN traffic.
This NLRI information is encoded using MP_REACH_NLRI and MP_UNREACH_NLRI attributes defined in [RFC4760]. When advertising FSv2 NLRI, the length of the Next-Hop Network Address MUST be set to 0. Upon reception, the Network Address in the Next-Hop field MUST be ignored.

Implementations wishing to exchange flow specification rules MUST use BGP’s Capability Advertisement facility to exchange the Multiprotocol Extension Capability Code (Code 1) as defined in [RFC4760], and indicate a capability for FSv1, FSv2 (Code TBD3), or both.

The AFI/SAFI NLRI for BGP Flow Specification version 2 (FSv2) has the format:

```
+-------------------------------+
|length (2 octets)             |
+-------------------------------+
| Sub-TLVs (variable)          |
| +-------------------------------+
| | order (4 octets)            |
| +-------------------------------+
| | identifier (4 octets)       |
| +-------------------------------+
| | type (2 octets)             |
| +-------------------------------+
| | length-Subtlv (2 octets)    |
| +-------------------------------+
| | value (variable)            |
| +-------------------------------+
```

Figure 3-1: FSv2 format

where:

* length: length of field including all SubTLVs in octets.

  - The combined lengths of any FSv2 NLRI in the MP_REACH_NLRI or MP_UNREACH_NLRI. The BGP NLRI length must be less than the packet size minus the other fields (BGP header, BGP Path Attributes, and NLRI).

* order: flow-specification global rule order number (4 octets).

* identifier: identifier for the rule (used for NM/Logging) (4 octets)
* type: contains a type for FSv2 TLV format of the NRLI (2 octets) which can be:
  - 0 - reserved,
  - 1 - IP Traffic Rules
  - 2- L2 traffic rules
  - 3- SFC Traffic rules
  - 4- SFC VPN Traffic rules
  - 5 - BGP/MPLS VPN IP Traffic Rules
  - 6 - BGP/MPLS VPN L2 Traffic Rules
* length-Subtlv: is the length of the value part of the Sub-TLV,
* value: value depends on the subTLV (see sections below).

3.1. IP header SubTLV (type=1)

The format of the IP header TLV value field is shown in figure 4. The AFI/SAFI field includes the AFI (2 octets), SAFI (1 octet). The AFI will be 1 (IPv4) or 2 (IPv6) and the SAFI will be TBD1 or, for the VPN case, TBD2. The IP header for the VPN case is specified in section 3.5.

```
+--------------------------+
| +--------------------------+  |
| | AFI/SAFI field (3)       |  |
| +--------------------------+  |
| | (subTLVs)+               |  |
| +==========================+  |
| +--------------------------+  |
```

Figure 3-2 - IP Header TLV

Where: AFI is 1 (IPv4) or 2 (IPv6) and SAFI is TBD1.

Each SubTLV has the format:
Figure 3-3  IP header SubTLV format

Where:

SubTLV type: component values are defined in the "Flow Specification Component types" registry for IPv4 and IPv6 by [RFC8955], [RFC8956], and [I-D.ietf-idr-flowspec-srv6]

length: length of SubTLV (varies depending on SubTLV type).

value: dependent on the subTLV

- For descriptions of value portions for components 1-13 see [RFC8955] and [RFC8956]. For component 14 see [I-D.ietf-idr-flowspec-srv6].

Many of the components use the operators [numeric_op] and [bitmask_op] defined in [RFC8955]

The list of valid SubTLV types appears in Table 2.

Table 2 IP SubTLV Types for IP Filters

<table>
<thead>
<tr>
<th>SubTLV-type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IP Destination prefix</td>
</tr>
<tr>
<td>2</td>
<td>IP Source prefix</td>
</tr>
<tr>
<td>3</td>
<td>IPv4 Protocol / IPv6 Upper Layer Protocol</td>
</tr>
<tr>
<td>4</td>
<td>Port</td>
</tr>
<tr>
<td>5</td>
<td>Destination Port</td>
</tr>
<tr>
<td>6</td>
<td>Source Port</td>
</tr>
<tr>
<td>7</td>
<td>ICMPv4 type / ICMPv6 type</td>
</tr>
<tr>
<td>8</td>
<td>ICMPv4 code / ICMPv6 code</td>
</tr>
<tr>
<td>9</td>
<td>TCP Flags</td>
</tr>
<tr>
<td>10</td>
<td>Packet length</td>
</tr>
<tr>
<td>11</td>
<td>DSCP (Differentiated Services Code Point)</td>
</tr>
<tr>
<td>12</td>
<td>Fragment</td>
</tr>
<tr>
<td>13</td>
<td>Flow Label</td>
</tr>
<tr>
<td>14</td>
<td>TTL</td>
</tr>
<tr>
<td>15</td>
<td>Parts of SID</td>
</tr>
<tr>
<td>16</td>
<td>MPLS Match 1: Label in Label stack</td>
</tr>
<tr>
<td>17</td>
<td>MPLS Match 2: EXP bits in top Label</td>
</tr>
</tbody>
</table>
Ordering within the TLV in FSv2: The transmission of SubTLVs within a flow specification rule MUST be sent ascending order by SubTLV type. If the SubTLV types are the same, then the value fields are compared using mechanisms defined in [RFC8955] and [RFC8956] and MUST be in ascending order. NLRIs having TLVs which do not follow the above ordering rules MUST be considered as malformed by a BGP FSv2 propagator. This rule prevents any ambiguities that arise from the multiple copies of the same NLRI from multiple BGP FSv2 propagators. A BGP implementation SHOULD treat such malformed NLRIs as "Treat-as-withdraw" [RFC7606].

See [RFC8955], [RFC8956], and [I-D.ietf-idr-flowspec-srv6]. for specific details.

3.1.1. IP Destination Prefix (type = 1)

IPv4 Name: IP Destination Prefix (reference: [RFC8955])
IPv6 Name: IPv6 Destination Prefix (reference: [RFC8956])

IPv4 length: Prefix length in bits
IPv4 value: IPv4 Prefix (variable length)
IPv6 length: length of value
IPv6 value: [offset (1 octet)] [pattern (variable)] [padding(variable)]

If IPv6 length = 0 and offset = 0, then component matches every address. Otherwise, length must be offset "less than" length "less than" 129 or component is malformed.

3.1.2. IP Source Prefix (type = 2)

IPv4 Name: IP Source Prefix (reference: [RFC8955])
IPv6 Name: IPv6 Source Prefix (reference: [RFC8956])

IPv4 length: Prefix length in bits
IPv4 value: Source IPv4 Prefix (variable length)
IPv6 length: length of value
IPv6 value: [offset (1 octet)] [pattern (variable)][padding(variable)]

If IPv6 length = 0 and offset = 0, then component matches every address. Otherwise, length must be offset < length < 129 or component is malformed.

3.1.3. IP Protocol (type = 3)

IPv4 Name: IP Protocol IP Source Prefix (reference: [RFC8955])
IPv6 Name: IPv6 Upper-Layer Protocol: (reference: [RFC8956])

IPv4 length: variable
IPv4 value: [numeric_op, value]+

IPv6 length: variable
IPv6 value: [numeric_op, value]+ where the value following each numeric_op is a single octet.

3.1.4. Port (type = 4)

IPv4/IPv6 Name: Port (reference: [RFC8955]), [RFC8956])

Filter defines: a set of port values to match either destination port or source port.

IPv4 length: variable
IPv4 value: [numeric_op, value]+

IPv6 length: variable
IPv6 value: [numeric_op, value]+ where the value following each numeric_op is a single octet.
Note-1: (from FSv1) In the presence of the port component (destination or source port), only a TCP (port 6) or UDP (port 17) packet can match the entire flow specification. If the packet is fragmented and this is not the first fragment, then the system may not be able to find the header. At this point, the FSv2 filter may fail to detect the correct flow. Similarly, if other IP options or the encapsulating security payload (ESP) is present, then the node may not be able to describe the transport header and the FSv2 filter may fail to detect the flow.

The restriction in note-1 comes from the inheritance of the FSv1 filter component for port. If better resolution is desired, a new FSv2 filter should be defined.

Note-2: FSv2 component only matches the first upper layer protocol value.

3.1.5. Destination Port (type = 5)
IPv4/IPv6 Name: Destination Port (reference: [RFC8955]), [RFC8956])
Filter defines: a list of match filters for destination port for TCP or UDP within a received packet
Length: variable
Component Value format: [numeric_op, value]+

3.1.6. Source Port (type = 6)
IPv4/IPv6 Name: Source Port (reference: [RFC8955]), [RFC8956])
Filter defines: a list of match filters for source port for TCP or UDP within a received packet
IPv4/IPv6 length: variable
IPv4/IPv6 value: [numeric_op, value]+

3.1.7. ICMP Type (type = 7)
IPv4: ICMP Type (reference: [RFC8955])
Filter defines: Defines: a list of match criteria for ICMPv4 type
IPv6: ICMPv6 Type (reference: [RFC8956])
Filter defines: a list of match criteria for ICMPv6 type.
IPv4/IPv6 length: variable
IPv4/IPv6 value: [numeric_op, value]+

3.1.8. ICMP Code (type = 8)
IPv4: ICMP Type (reference: [RFC8955])
Filter defines: a list of match criteria for ICMPv4 code.
IPv6: ICMPv6 Type (reference: [RFC8956])
Filter defines: a list of match criteria for ICMPv6 code.
IPv4/IPv6 length: variable
IPv4/IPv6 value: [numeric_op, value]+

3.1.9. TCP Flags (type = 9)
IPv4/IPv6: TCP Flags Code (reference: [RFC8955])
Filter defines: a list of match criteria for TCP Control bits
IPv4/IPv6 length: variable
IPv4/IPv6 value: [bitmask_op, value]+

Note: a 2 octets bitmask match is always used for TCP-Flags

3.1.10. Packet length (type = 10 (0x0A))
IPv4/IPv6: Packet Length (reference: [RFC8955], [RFC8956])
Filter defines: a list of match criteria for length of packet (excluding L2 header but including IP header).
IPv4/IPv6 length: variable
IPv4/IPv6 value: [numeric_op, value]+

Note: [RFC8955] uses either 1 or 2 octet values.
3.1.11. DSCP (Differentiated Services Code Point) (type = 11 (0x0B))

IPv4/IPv6: DSCP Code (reference: [RFC8955], [RFC8956])

Filter defines: a list of match criteria for DSCP code values to match the 6-bit DSCP field.

IPv4/IPv6 length: variable

IPv4/IPv6 value: [numeric_op, value]+

Note: This component uses the Numeric Operator (numeric_op) described in [RFC8955] in section 4.2.1.1. Type 11 component values MUST be encoded as single octet (numeric_op len=00).

The six least significant bits contain the DSCP value. All other bits SHOULD be treated as 0.

3.1.12. Fragment (type = 12 (0x0C))

IPv4/IPv6: Fragment (reference: [RFC8955], [RFC8956])

Filter defines: a list of match criteria for specific IP fragments.

Length: variable

Component Value format: [bitmask_op, value]+

Bitmask values are:

```
  0 1 2 3 4 5 6 7
+-----------------------
| 0 | 0 | 0 | 0 | LF | FF | IsF | DF |
+-----------------------
```

Figure 3-4

Where:

DF (don’t fragment): match if IP header flags bit 1 (DF) is 1.

IsF(is a fragment other than first): match if IP header fragment offset is not 0.

FF (First Fragment): Match if [RFC0791] IP Header Fragment offset is zero and Flags Bit-2 (MF) is 1.
LF (last Fragment): Match if [RFC7091] IP header Fragment is not 0
And Flags bit-2 (MF) is 0

0: MUST be sent in NLRI encoding as 0, and MUST be ignored during reception.

3.1.13. Flow Label(type = 13 (0x0D))

IPv4/IPv6: Fragment (reference: [RFC8956])

Filter defines: a list of match criteria for 20-bit Flow Label in the IPv6 header field.

Length: variable

Component Value format: [numeric_op, value]+

3.1.14. TTL (type=14 (0x0E))

TTL: Defines matches for 8-bit TTL field in IP header

Encoding: <[numeric_op, value]+>

where: value is a 1 octet value for TTL.

ordering: by full value of number_op concatenated with value

conflict: none

reference: draft-bergeon-flowspec-ttl-match-00.txt

3.1.15. Parts of SID (type = 15 (0xF))


Filter defines: a list of match bit match criteria for some combinations of the LOC (location), FUNCT (function) and ARG (arguments) fields in the SID or or whole SID.

Length: variable

Component Value format: [type, LOC-Len, FUNCT-Len, ARG-Len, [op, value]+]

where:
* type (1 octet): This indicates the new component type (TBD1, which is to be assigned by IANA).

* LOC-Len (1 octet): This indicates the length in bits of LOC in SID.

* FUNCT-Len (1 octet): This indicates the length in bits of FUNCT in SID.

* ARG-Len (1 octet): This indicates the length in bits of ARG in SID.

* [op, value]+: This contains a list of {operator, value} pairs that are used to match some parts of SID.

The total of three lengths (i.e., LOC length + FUNCT length + ARG length) MUST NOT be greater than 128. If it is greater than 128, an error occurs and it is treated as a withdrawal [RFC7606] and [RFC4760].

The operator (op) byte is encoded as:

```
+---+---+---+---+---+---+---+---+
| e | a | field type|lt |gt |eq |
+---+---+---+---+---+---+---+
```

Figure 3-5

where:

where the behavior of each operator bit has clear similarity with that of [RFC8955]’s Numeric Operator field.

e (end-of-list bit): Set in the last {op, value} pair in the sequence.

a - AND bit: If unset, the previous term is logically ORed with the current one. If set, the operation is a logical AND. It should be unset in the first operator byte of a sequence. The AND operator has higher priority than OR for the purposes of evaluating logical expressions.

field type:

- 000: SID’s LOC
- 001: SID’s FUNCT
- 010: SID’s ARG
- 011: SID’s LOC:FUNCT (the concatenation of the LOC and FUNCTION fields)
- 100: SID’s FUNCT:ARG (the concatenation of the FUNCTION and ARG fields)
- 101: SID’s LOC:FUNCT:ARG (the concatenation of the FUNCTION and ARG fields)

Note: For an unknown field type, Error Handling is to "treat as withdrawal" [RFC7606] and [RFC4760].

lt: less than comparison between data’ and value’.

gt: greater than comparison between data’ and value’.

eq: equality between data’ and value’.

The data’ and value’ used in lt, gt and eq are indicated by the field type in an operator and the value field following the operator.

The length of the value field depends on the field type and is the length of the SID parts being matched (see Table 3, Figure 3-6) in bytes, rounded up if that length is not a multiple of 8.

Table 3 - SID Parts fields

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID’s LOC</td>
<td>value of LOC bits</td>
</tr>
<tr>
<td>SID’s FUNCT</td>
<td>value of FUNCT bits</td>
</tr>
<tr>
<td>SID’s ARG</td>
<td>value of ARG bits</td>
</tr>
<tr>
<td>SID’s LOC:FUNCT</td>
<td>value of LOC:FUNCT bits</td>
</tr>
<tr>
<td>SID’s FUNCT:ARG</td>
<td>value of FUNCT:ARG bits</td>
</tr>
<tr>
<td>SID’s LOC:FUNCT:ARG</td>
<td>value of LOC:FUNCT:ARG bits</td>
</tr>
</tbody>
</table>
3.1.16. MPLS Label Match1 (type=16, 0x10)

Function: This match1 applies to MPLS Label field on the label stack.

reference: [I-D.ietf-idr-flowspec-mpls-match]

Encoding: <type(1 octet), length(1 octet), [operator,value]+>.

It contains a set of (operator, value) pairs that are used for the matching filter.

The operator byte is encoded as:

```
+---+---+---+---+---+---+---+---+
| e | a | i |  pos |   Resv    |
+---+---+---+---+---+---+---+---+
```

Figure 3-7

where:

e - end of list bit: Set in the last (op, value) pair in the list.

a - AND bit: If unset, the previous term is logically ORed with the current one. If set, the operation is a logical AND. It should be unset in the first operator byte of a sequence. The AND operator has higher priority than OR for the purposes of evaluating logical expressions.

i - before bit: If unset, apply matching filter before MPLS label
data plane action; if set, apply matching filter after MPLS label data plane action.

pos - the label position indication bits: whose meaning for various values is shown below:

00: any position on the label stack - the presented label value is used to match any label on the label stack. When applying it, at least one label on the stack MUST match the value.

01: top label indication - the presented label value MUST be used to match the top label on the label stack.

10: bottom label indication - the presented label value MUST match the bottom label on the label stack. When it is clear, the present label value can match to any label on the label stack.

11: reserved value - This value is reserved and MUST not be sent in the packet.

The value field is encoded as:

```
  1                   2
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
 +-----------------------------------------------+
 |                Label                         |
 +-----------------------------------------------+
```

reference:

3.1.17. MPLS Label Match 2: Experimental bits match on top label (Type=17 (0x11))

Function: MPLS Match2 applies to MPLS Label experiment bits (EXP) on the top label in the label stack.

reference: [I-D.ietf-idr-flowspec-mpls-match]

Encoding: <type (1 octet), [op, value]>

- [op, value] - Defines a list of (operation, value) pairs used to match 3-bit exp field on the top label of packets [RFC3032].
3.2. Encoding of FSv2 Actions (type=2)

The FSv2 actions may be sent in an Extended Community or a Wide Community.

The Extended Community encodes the Flow Specification actions in the Extended Community format [RFC4360].

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Type high    |  Type low(*)  |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+          Value (6 octets)     |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3-9

The Wide Community definition for FSv2 actions is as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type=1            |    Flags  |C|T|   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             |<sequence of FSv2-Action-TLV>+ |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3-10

where FSv2-Action-TLV is defined as:

```
FSv2-Action-TLV
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    action order                  |   Chain-ID                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          <action-subTLVs>                                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3-11

Where action-SubTLVs have the format:
action-SubTLVs

<table>
<thead>
<tr>
<th>SubTLV type (2 octet)</th>
<th>length type (2 octet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>value (variable)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3-12

where:

action-order: is the user defined order of the action within the list

chain ID: is a 2-byte identifier for an action chain

length - is the length of the TLV

value - contains a sequence of action SubTLVs.

Each Action SubTLV has the format:

<table>
<thead>
<tr>
<th>SubTLV type (2 octet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>length (2 octet)</td>
</tr>
<tr>
<td>value (variable)</td>
</tr>
</tbody>
</table>

Figure 3-14

Where:

* SubTLV type: values are action type values shown in Table 4 below.

* length: is the length of the action SubTLV

* Value is specific to the SubTLV
Table 4  FSv2 Action types

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>reserved</td>
</tr>
<tr>
<td>01</td>
<td>ACO: action chain operation</td>
</tr>
<tr>
<td>02</td>
<td>TAIS: traffic actions per interface group</td>
</tr>
<tr>
<td>06</td>
<td>TRB: traffic rate limited by bytes</td>
</tr>
<tr>
<td>07</td>
<td>TA: traffic action (terminal/sample)</td>
</tr>
<tr>
<td>08</td>
<td>RDIP: Redirect IPv4</td>
</tr>
<tr>
<td>09</td>
<td>TM: mark DSCP value</td>
</tr>
<tr>
<td>10</td>
<td>TBA (to be assigned)</td>
</tr>
<tr>
<td>11</td>
<td>TBA (to be assigned)</td>
</tr>
<tr>
<td>12</td>
<td>TRP: traffic rate limited by packets</td>
</tr>
<tr>
<td>13</td>
<td>RDIPv6: redirect to IPv6</td>
</tr>
<tr>
<td>14</td>
<td>TISFC: SFC Classifier Info (moved from OD to OE)</td>
</tr>
<tr>
<td>15</td>
<td>RDIID: redirect to Indirection-id (move from 0x00)</td>
</tr>
<tr>
<td>16</td>
<td>MPLSLA: MPLS label action</td>
</tr>
<tr>
<td>17-21</td>
<td>TBA (to be assigned)</td>
</tr>
<tr>
<td>22</td>
<td>VLAN: VLAN-Action (0x16)[draft-ietf-idr-flowspec-12vpn-17]</td>
</tr>
<tr>
<td>23</td>
<td>TPID: TPID-Action (0x17)[draft-ietf-idr-flowspec-12vpn-17]</td>
</tr>
<tr>
<td>24-254</td>
<td>TBA (to be assigned)</td>
</tr>
<tr>
<td>255</td>
<td>reserved</td>
</tr>
</tbody>
</table>

Figure 3-15

Ordering of actions within a rule:

The actions are first stored in user-defined order. If multiple actions exist for a single action order value, then the actions will be ordered by action type followed by value.

Action specifications must include descriptions of order comparison for the values within the action.

3.2.1. Action Chain operation (ACO) (1, 0x01)

SubTLV: 0x01

Length: variable

Value:

AC-failure-type - byte that determines the action on failure
AC-failure-value - variable depending on AC-failure-type.
Actions may succeed or fail and an Action chain must deal with it. The default value stored for an action chain that does not have this action chain is "stop on failure".

where:

AC-Failure types are:
- 0x00 - default - stop on failure
- 0x01 - continue on failure (best effort on actions)
- 0x02 - conditional stop on failure - depending on AC-Failure-value
- 0x03 - rollback - do all or nothing - depending in AC-Failure-value

AC-Failure values: TBD

Interactions with other actions: Interactions with all other Actions

Ordering within Action type: By AC-Failure type

3.2.2. Traffic Actions per interface set (TAIS) (2, 0x02)

SubTLV: 0x02

Length: 8 octets (6 in extended community)

Value field: [4-octet-AS] [GroupID 2-octet] [action 2-octet]

where:

Group-ID: identifier for group in 2 octets (14 lower bits)
- Note: Extended Community format will have 2 bits for action.

Action: determines inbound or outbound action where:
- Outbound (0x1): FSv2 rule MUST be applied in outbound Direction to interface set identified by Group-ID.
- Inbound (0x2): FSv2 rule MUST be applied in inbound Direction to interface set identified by Group-ID.

Value ordering: AS, then Group ID, then Action bytes.
Conflict: with any bi-direction action such as

1. traffic rate limited by bytes, or
2. traffic rate limited by packets.

Reference: [I-D.ietf-idr-flowspec-interfaceset]

3.2.3. Traffic rate limited by bytes (TRB) (6, 0x06)

SubTLV: 0x06
Length: 8 octets
Value field: [4-octet-AS] [float (4 bytes)]

where:

[4-octet-AS]: 4 byte AS number
- If FSv1 passes the lower 2 bytes of 4 byte AS number, use [TBD6] as higher 2 bytes to identify.
- Open issue: TBD6 can be either be chosen to match the common 2-byte to 4-byte or a unique value. Feedback is needed from WG and authors.

Float: maximum byte rate in IEEE 32-bit floating point [IEEE.754.19895 format] in bytes per second.
- A value of 0 should result in all traffic for the particular flow to be discarded.
- On encoding the traffic-rate-packets MUST NOT be negative.
- On decoding, negative values MUST BE treated as zero (discard all traffic).

Value ordering: AS then float value

Action Conflict: traffic-rate-packets
reference: [RFC8955]

3.2.4. Traffic Action (TA) (7, 0x07)

SubTLV: 0x07
Length: 1

Value field: [1-octet action]

where the traffic action values are:

1 = Terminal flow specification action
2 = Sample - enables sampling and logging
3 = Terminal action + sample

Value ordering: By traffic action values

Conflicts/Interactions: duplication of packets also occurs in:

Redirect to IPv4 (action 0x08),
Redirect to IPv6 (action 0x0D (13)),
Redirect to SFC (action 0x0E (14))
Redirect to Indirection-ID (action 0xF (15))

3.2.5. Redirect to IPv4 (RDIPv4)(8, 0x08)

SubTLV: 0x08

Length: 12 octets

Value field:

[4-byte-AS] [IPv4 address (4 octets) [ID (4 octets)] [Flag (1 octet)]

where:

4-octet-AS - is a 4-byte AS in a Route Target
IPv4 address - is an IP Address in RT
ID - the 4-octet value set by user
Flag is 1 octet value with the following definitions:
- 0 - reserved
- 1 - copy and redirect copy
Value ordering: 4-octet AS, then IP address, then ID (lowest to highest) with:

No AS specified uses AS value of zero.

No IP specified uses IP value of zero.

No ID specified uses ID value of zero.

Conflicts/Interactions: Any redirection or traffic sampling found in:

Traffic Action (action 0x07),

Redirect to IPv6 (action 0x0D (13)),

Redirect to SFC (action 0x0E (14))

Redirect to Indirection-ID (action 0xF (15))

reference: [RFC8955], draft-ietf-idr-flowspec-ip-02.txt

3.2.6. Traffic marking (TM) (9, 0x09)

SubTLV: 0x09

Length: 1 octet

Value: DSCP field with the 2 left most bits zero

The DSCP field format is:

```
0 1 2 3 4 5 6 7
+--------+--------+
| R | R | DSCP bits |
+--------+--------+
```

Figure 3-16

where:

R - reserved bits (set to zero to send, ignored upon reception and set to zero.

DSCP - 6 bits of DSCP values

Ordering within Value: Based on DSCP value
Conflicts: none
reference: [RFC8955]

3.2.7. Traffic rate limited by packets (TRP) (12, 0xC)

SubTLV: 12 (0xC)
Length: 8
Value field: [4-octet-AS] [float (4 octet)]

Where:

4-octet AS - is the AS setting this value

Float - specifies maximum rate in IEEE 32-bit format
[IEEE.754.185] in packets per second.

- A traffic rate of zero should result in all packets being
discard.

- On encoding the traffic-rate-packets MUST NOT be negative.

- On decoding, negative values MUST BE treated as zero (discard
all traffic).

Ordering within Value: Based on DSCP value

Conflicts: Traffic rate limited by bytes (0x06)
reference: [RFC8955]

3.2.8. Traffic redirect to IPv6 (RDIPv6) (13, 0xD)

SubTLV: 13 (0xD)
Length: 24 octets

Value field: [4-octet-as] [IPv6-address (16 octets)] [local
administrator (2 octets)] [Flag (1 octets)]

where:

4-octet-AS - is the AS requesting action in 4-byte AS format,
IPv6-address - is the redirection IPv6 address
Local administrator - 2 bytes assigned by network administrator.

lag (1 octet) with the following definitions:
- 0 - reserved
- 1 - copy and redirect copy

Ordering within Value: AS, then IPv6, the flag (low to high)
Conflicts/Interactions: Any redirection or traffic sampling found in:
- Traffic Action (action 0x07),
- Redirect to IPv4 (action 0x08 (8)),
- Redirect to SFC (action 0x0E (14))
- Redirect to Indirection-ID (action 0xF (15))

3.2.9. Traffic insertion in SFC (TISFC) (14, 0xE)

SubTLV: 14 (0xE)

Note: replace IANA 0xD FSv1 with FSv2 0xE.

Length: 6 octets

Value field: [SPI (3 octets)][SI (1 octet)][SFT (2 octet)]
where:
- SPI - is the service path identifier
- SI - is the service index
- SFT - is the service function type.

Value ordering: SPI, then SI, then SFT (lowest to highest)
Conflicts/Interactions: Any redirection or traffic sampling found in:
- Traffic Action (action 0x07),
- Redirect to IPv4 (action 0x08 (8)),
Redirect to IPv6 (action 0xOD (13))

Redirect to Indirection-ID (action 0xF (15))

Reference: [RFC9015]

3.2.10. Flow Specification Redirect to Indirection-ID (RDIID) (15, 0x0F)

SubTLV: 15 (0x0F)

note: current value is 0x00 for FSv1

Length: 6 octets

Value field:

[Flags (1 octet)] [ID-Type (1 octet)][Generalized-ID (4 octets)]

Figure 3-17

where:

Flags: are defined as:

- [S-ID]: sequence number for indirection IDs (3 bits).
  - Value of zero means sequence is not set and all other S-ID values should be ignored
- [C] - copy packets matching this ID

ID-Type: type of indirection ID with following values:

- 0 - localized ID
- 1 - Node with SID/index in MPLS SR
- 2 - Node with SID/label in MPLS SR
- 3 - Node with Binding Segment ID with SID/Index
- 4 - Node with Binding Segment ID with SID/Label
- 5 - Tunnel ID

Generalized-ID (G-ID): indirection value
Value Ordering: first indirection ID, then Generalized ID

Action Value ordering: ID-Type by value (lowest to highest)

Conflicts/Interactions: Any redirection or traffic sampling found in:
  Traffic Action (action 0x07),
  Redirect to IPv4 (action 0x08 (8)),
  Redirect to IPv6 (action 0x0D, (13)
  Redirect to SFC (action 0x0E (14))

reference: [I-D.ietf-idr-flowspec-path-redirect]

3.2.11. MPLS Label Action (MPLSLA) (16, 0x10)

Function: MPLS Label actions

SubTLV: 16 (0x10)

Length: 6 octets

Value:

  [action (1 octet)
   [order (1 octet)
   [Label Stack Entry (4 octets)]

where Action:

<table>
<thead>
<tr>
<th>Action</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Push the MPLS tag</td>
</tr>
<tr>
<td>1</td>
<td>Pop the outermost MPLS tag in the packet</td>
</tr>
<tr>
<td>2</td>
<td>Swap the MPLS tag with the outermost MPLS tag in the packet</td>
</tr>
<tr>
<td>3˜15</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Figure 3-18
Action Value ordering: ID-Type, then value (lowest to highest)

Value Ordering: order, action, label, Exp

Conflicts/Interactions: Any redirection for IP before MPLS
- Traffic Action (action 0x07),
- Redirect to IPv4 (action 0x08 (8)),
- Redirect to IPv6 (action 0x0D, (13))
- Redirect to SFC (action 0x0E (14))

reference: [I-D.ietf-idr-bgp-flowspec-label]

3.2.12. VLAN action (VLAN) (22, 0x16)

Function: Rewrite inner or outer VLAN header

SubTLV: 22 (0x16)
Length: 6 octets

Value:
- [Rewrite-actions (2 octets)]
- [vlan-PCP-DE-1 (2 octets)]
- [vlan-PCP-DE-2 [2 octets]]

where:
Rewrite-actions - is as follows:
Figure 3-20

PO1: Pop action. If the PO1 flag is one, it indicates the outermost VLAN should be removed.

PU1: Push action. If PU1 is one, it indicates VLAN ID1 will be added, the associated Priority Code Point (PCP) and Drop Eligibility Indicator (DEI) are PCP1 and DE1.

SW1: Swap action. If the SW1 flag is one, it indicates the outer VLAN and inner VLAN should be swapped.

PO2: Pop action. If the PO2 flag is one, it indicates the outermost VLAN should be removed.

PU2: Push action. If PU2 is one, it indicates VLAN ID2 will be added, the associated PCP and DEI are PCP2 and DE2.

SW2: Swap action. If the SW2 flag is one, it indicates the outer VLAN and inner VLAN should be swapped.

RI1 and RI2: Rewrite inner VLAN action. If the RIx flag is one where "x" is "1" or "2"). it indicates the inner VLAN should be replaced by a new VLAN where the new VLAN is VLAN IDx and the associated PCP and DEI are PCPx and DEx. If the VLAN IDx is 0, the action is to only modify the PCP and DEI value of the inner VLAN.

RO1 and RO2: Rewrite outer VLAN action. If the ROx flag is one (where "x" is "1" or "2"). it indicates the outer VLAN should be replaced by a new VLAN where the new VLAN is VLAN IDx and the associated PCP and DEI are PCPx and DEx. If the VLAN IDx is 0, the action is to only modify the PCP and DEI value of the outer VLAN.

Resv: Reserved for future use. MUST be sent as zero and ignored on receipt.

Value ordering: rewrite-actions, VLAN1, VLAN2, PCP-DE1, PCP-DE2

Conflicts: TIPD Action

reference: [I-D.ietf-idr-flowspec-l2vpn]
3.2.13. TPID action (TPID) (23, 0x17)

Function: Replace Inner or outer TP

SubTLV: 23 (0x17)

Length: 6 octets

Value:

[Rewrite-actions (2 octets)]

[TP-ID-1 (2 octets)]

[TP-ID-2 (2 octets)]

Where: rewrite-actions are bitmask (2 octets) with 2 actions as follows:

<table>
<thead>
<tr>
<th>TI</th>
<th>TO</th>
<th>Resv</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3-21

TI: Mapping inner Tag Protocol (TP) ID (typically a VLAN) action. If the TI flag is one, it indicates the inner TP ID should be replaced by a new TP ID, the new TP ID is TP ID1.

TO: Mapping outer TP ID action. If the TO flag is one, it indicates the outer TP ID should be replaced by a new TP ID, the new TP ID is TP ID2.

Resv: Reserved for future use. MUST be sent as zero and ignored on receipt.

Value Ordering: rewrite-actions, TP-ID-1, TP-ID-2

Conflicts: VLAN action

reference: [I-D.ietf-idr-flowspec-l2vpn]

3.3. Extended Community vs. Action SubTLV formats

The SubTLV format is used for the Wide communities and for the action subTLVs in the NLRI.
<table>
<thead>
<tr>
<th>Sub-TLV</th>
<th>Action Action SubTLV</th>
<th>Extended Community</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>type</td>
<td>Name</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>ACO</td>
</tr>
<tr>
<td></td>
<td>length:variable</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>TAIS</td>
</tr>
<tr>
<td></td>
<td>length:8</td>
<td>length: 6</td>
</tr>
<tr>
<td></td>
<td>[4-octet-as]</td>
<td>[4-octet-AS]</td>
</tr>
<tr>
<td></td>
<td>[group-3-octet]</td>
<td>[flags-group]</td>
</tr>
<tr>
<td></td>
<td>[flags-1-octet]</td>
<td>(2 octets)</td>
</tr>
<tr>
<td>3-5</td>
<td>reserved</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>TRB</td>
</tr>
<tr>
<td></td>
<td>length:8</td>
<td>length: 6 octets</td>
</tr>
<tr>
<td></td>
<td>[4-byte-AS]</td>
<td>[2-byte-AS]</td>
</tr>
<tr>
<td></td>
<td>[float (4 octets)]</td>
<td>[float (4 octets)]</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>TA</td>
</tr>
<tr>
<td></td>
<td>length:1</td>
<td>length: 6 octets</td>
</tr>
<tr>
<td></td>
<td>flags: (1 octet)</td>
<td>flags (6 octets)</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>RDIPv4</td>
</tr>
<tr>
<td></td>
<td>length: 12</td>
<td>length: 6 octets</td>
</tr>
<tr>
<td></td>
<td>[4-byte-AS]</td>
<td>[AS-2-octets]</td>
</tr>
<tr>
<td></td>
<td>[IPv4-address]</td>
<td>[IPv4 address]</td>
</tr>
<tr>
<td></td>
<td>type:8108</td>
<td>length: 6 octets</td>
</tr>
<tr>
<td></td>
<td>[IPv4 address]</td>
<td>[ID-2 octets]</td>
</tr>
<tr>
<td></td>
<td>type:8208</td>
<td>length: 6 octets</td>
</tr>
<tr>
<td></td>
<td>[AS-4-octets]</td>
<td>[ID-2-octets]</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>TM</td>
</tr>
<tr>
<td></td>
<td>length:1</td>
<td>length: 6 octets</td>
</tr>
<tr>
<td></td>
<td>DSCP: 1 octet</td>
<td>DSCP: 1 octet</td>
</tr>
<tr>
<td>Sub-TLV</td>
<td>Action</td>
<td>SubTLV</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>10</td>
<td>type:10 (0X0A)</td>
<td>TBA</td>
</tr>
<tr>
<td>11</td>
<td>type:11 (0x0B)</td>
<td>TBA</td>
</tr>
<tr>
<td>12</td>
<td>TRP</td>
<td>type:12 (0x0C)</td>
</tr>
<tr>
<td></td>
<td>length: 8 octets</td>
<td>length: 6 octets</td>
</tr>
<tr>
<td></td>
<td>[4-byte-AS]</td>
<td>[2-byte-AS]</td>
</tr>
<tr>
<td></td>
<td>[float-4-octet]</td>
<td>[float-4-octet]</td>
</tr>
<tr>
<td>13</td>
<td>RDIPv6</td>
<td>type:13 (0x0D)</td>
</tr>
<tr>
<td></td>
<td>length:22</td>
<td>length: 18 octets</td>
</tr>
<tr>
<td></td>
<td>[4-byte-AS]</td>
<td>[IPv6-address (16)]</td>
</tr>
<tr>
<td></td>
<td>[IPv6-address (16)]</td>
<td>[local-admin (2)]</td>
</tr>
<tr>
<td></td>
<td>[local-admin (2)]</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>TISFC</td>
<td>type:14 (0x0E)</td>
</tr>
<tr>
<td></td>
<td>length:6</td>
<td>length:6</td>
</tr>
<tr>
<td></td>
<td>SPI (3 octets)</td>
<td>SPI (3 octets)</td>
</tr>
<tr>
<td></td>
<td>SI (1 octet)</td>
<td>SI (1 octet)</td>
</tr>
<tr>
<td></td>
<td>SFT (2 octets)</td>
<td>SFT (2 octets)</td>
</tr>
<tr>
<td>15</td>
<td>RDIID</td>
<td>type:15 (0x0F)</td>
</tr>
<tr>
<td></td>
<td>length: 6</td>
<td>length 6</td>
</tr>
<tr>
<td></td>
<td>flags (1)</td>
<td>flags (1)</td>
</tr>
<tr>
<td></td>
<td>ID-type (1)</td>
<td>ID type (1)</td>
</tr>
<tr>
<td></td>
<td>G-ID (4 octets)</td>
<td>G-ID (4-octets)</td>
</tr>
<tr>
<td>16</td>
<td>MPLSLA</td>
<td>type:16 (0x10)</td>
</tr>
<tr>
<td>16-21</td>
<td>TBA</td>
<td>-</td>
</tr>
<tr>
<td>22</td>
<td>VLAN</td>
<td>type:22 (0x16)</td>
</tr>
<tr>
<td></td>
<td>length:6</td>
<td>length:6</td>
</tr>
<tr>
<td></td>
<td>[rewrite-action(2)]</td>
<td>[rewrite-actions (2)]</td>
</tr>
<tr>
<td></td>
<td>[vlan-pcp-de-1 (2)]</td>
<td>[vlan-pcp-de-1 (2)]</td>
</tr>
<tr>
<td></td>
<td>[vlan-pcp-de-2 (2)]</td>
<td>[vlan-pcp-de-2 (2)]</td>
</tr>
<tr>
<td>23</td>
<td>TPID</td>
<td>type:23 (0x17)</td>
</tr>
<tr>
<td></td>
<td>length:6</td>
<td>length:6</td>
</tr>
</tbody>
</table>
3.4. L2 Traffic Rules

The format of the L2 header TLV value field is shown in Figure 3-22. The AFI/SAFI field includes the AFI (2 octets), SAFI (1 octet).

```
+-------------------------------+
<table>
<thead>
<tr>
<th>AFI/SAFI field (3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L3 AFI (2)</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>L2 filter length (2)</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>(SubTLVs)+ (L2 then L3)</td>
</tr>
</tbody>
</table>
+-------------------------------+
```

Figure 3-22 - L2 Header TLV value

Where:

- AFI/SAFI field has AFI is 6 (IEEE 802) and SAFI is TBD1.

- L3 AFI is zero if the filter test only L2 fields, otherwise it is 0 or 2 depending on whether the filter L3 tests after the L2 header are for IPv4 or IPv6.

- L2 filter length is the length of the L2 SubTLVs in bytes. These are followed by the L3 SubTLVs if the L3 AFI field is non-zero.

Each L2 SubTLV has the format shown in Figure 3-23. (The L3 SubTLVs are as defined in Section 4.1.)

```
+------------------------------------------+
<table>
<thead>
<tr>
<th>SubTLV type (1 octet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>length (1 octet)</td>
</tr>
</tbody>
</table>
+------------------------------------------|
|   value (variable)                       |
+------------------------------------------+
```

Figure 3-23
SubTLV type: A component type value defined in the "L2 Flow Specification Component Types" registry for L2 by [draft-ietf-idr-flowspec-l2vpn].

Where the SubTLVs have the following component types:

Component Types Table

<table>
<thead>
<tr>
<th>Component type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EtherType</td>
</tr>
<tr>
<td>2</td>
<td>Source MAC</td>
</tr>
<tr>
<td>3</td>
<td>Destination MAC</td>
</tr>
<tr>
<td>4</td>
<td>DSAP (destination service access point)</td>
</tr>
<tr>
<td>5</td>
<td>SSAP (source service access point)</td>
</tr>
<tr>
<td>6</td>
<td>control field in LLC</td>
</tr>
<tr>
<td>7</td>
<td>SNAP</td>
</tr>
<tr>
<td>8</td>
<td>VLAN ID</td>
</tr>
<tr>
<td>9</td>
<td>VPAN PCP</td>
</tr>
<tr>
<td>10</td>
<td>Inner VLAN ID</td>
</tr>
<tr>
<td>11</td>
<td>Inner VLAN PCP</td>
</tr>
<tr>
<td>12</td>
<td>VLAN DEI</td>
</tr>
<tr>
<td>13</td>
<td>VLAN DEI</td>
</tr>
<tr>
<td>14</td>
<td>Source MAC special bits</td>
</tr>
<tr>
<td>15</td>
<td>Destination MAC special bits</td>
</tr>
</tbody>
</table>

Table 4  L2 VPN components

See [I-D.ietf-idr-flowspec-l2vpn] for the details on the format and value fields for each component.

Value ordering: Ordering of L2 FSv2 rules will be by user-defined order of the rule. For FSv2 filters within the same rule, the ordering will be by component number and then by value within the component. See [I-D.ietf-idr-flowspec-l2vpn] for the ordering of the values within the component.

L2 VPN filtering using SAFI TBD2 is specified in section 3.6.

reference: [I-D.ietf-idr-flowspec-l2vpn]

3.5. SFC Traffic Rules

The FSv2 filters allow for filtering of the SFC NLRI family of routes. The traffic NLRIs filtered are from SFC AFI/SAFI (AFI = 31, SAFI=9).
The FSv2 filters provide this filtering with SFC AFI (AFI=31) and SAFI for FSv2 filters (SAFI = TB1).

```
+--------------------------------------+
| +---------------------------------+  |
| | Tunneled AFI/SAFI field         |  |
| +---------------------------------+  |
| |                                 |  |
| | <subTLVs>+                      |  |
| +---------------------------------+  |
+--------------------------------------+
```

Figure 3-24

Each SubTLV has the format:

```
+-------------------------------+
|  SubTLV type (1 octet)        |
+-------------------------------+
|  length (1 octet)             |
| + ------------------------------+
|  value (variable)             |
+-------------------------------+
```

Figure 3-25 Tunneled SubTLV format

The components listed are:

1 = SFIR RD Type (types 1, 2, 3)
2 = SFIR RD Value
3 = SFIR Pool ID
4 = SFIR MPLS context/label
5 = SFPR SPI
6 = SPF attribute fields

Table 6 SFC Filter types

Ordering is by: User-defined rule order, component number, and then value within component.

reference: [RFC9015], [TBD]
3.6. BGP/MPLS VPN IP Traffic Rules

The format of the match filter for BGP/MPLS VPN IP traffic is very similar to the format for non-VPN IP traffic as defined in Section 3.1 except that the SAFI is TBD2 and the initial NLRI header has an 8-byte Route Distinguisher added to it as shown in Figure 3-26. The SubTLV format and filter components formats remain the same.

```
+-------------------------------+
| +--------------------------+  |
| | AFI/SAFI field (3)       |  |
| +--------------------------+  |
| | Route Distinguisher (8)  |  |
| +--------------------------+  |
| | (subTLVs)+               |  |
| +-------------------------------+
```

Figure 3-26: VPN IP Filter Header

3.7. BGP/MPLS VPN L2 Traffic Rules

The format of the match filter for BGP/MPLS VPN IP traffic is very similar to the format for non-VPN L2 traffic as defined in Section 3.4 except that the SAFI is TBD2 and the initial NLRI header has an 8-byte Route Distinguisher added to it right after the AFI/SAFI as shown in Figure 3-27. The SubTLV format and filter components formats remain the same.

```
+-------------------------------+
| +--------------------------+  |
| | AFI/SAFI field (3)       |  |
| +--------------------------+  |
| | Route Distinguisher (8)  |  |
| +--------------------------+  |
| | L3 AFI (2)               |  |
| +--------------------------+  |
| | L2 filter length (2)     |  |
| +--------------------------+  |
| | (subTLVs)+               |  |
| +-------------------------------+
```

Figure 3-27: VPN L2 Filter Header
3.8. Encoding of Actions passed in Wide Communities

The BGP FSv2 actions are passed in a Wide Community attribute with a BGP Wide Community container (type 01) [I-D.ietf-idr-wide-bgp-communities] with community of FSv2 Actions (TBD4) and Wide Community attributes of Target TLV, Exclude TLVs, and Parameter TLVs. The Parameter MUST contains an FSv2 Atom which contains a sequence of Action TLVs.

BGP Wide Community Container with FSv2 actions

```
+----------------------------+
| Community: FSv2-actions    |
| (community value = TBD4)   |
+----------------------------+
| Source AS number           |
+----------------------------+
| Context AS number          |
+----------------------------+
| Target or Exclude TLVs     +
| (optional)                 |
+----------------------------+
| Parameter TLV with         |
| FSv2 atom                  |
+----------------------------+
```

Figure 3-28 - BGP

```
+----------------------------+
| FSv2 Actions atom-id       |
+----------------------------+
| length (2 octets)          |
+----------------------------+
| <Action-Sub-TLVs>+         |
+----------------------------+
```

Figure 3-29 - Flow Specification with IDs for Wide Community Actions

where:

Atom-id: (TBD5)

length: variable depending on SubTLVS

Action Sub-TLVs as defined above
4. Validation of FSv2 NLRI

The validation of FSv2 NLRI adheres to the combination of rules for general BGP FSv1 NLRI found in [RFC8955], [RFC8956], [RFC9117], and the specific additions made for SFC NLRI [RFC9015], and L2VPN NLRI [I-D.ietf-idr-flowspec-l2vpn].

To provide clarity, the full validation process for flow specification routes (FSv1 or FSv2) is described in this section rather than simply referring to the relevant portions of these RFCs. Validation only occurs after BGP UPDATE message reception and the FSv2 NLRI and the path attributes relating to FSv2 (Extended community and Wide Community) have been determined to be well-formed. Any MALFORMED FSv2 NRLI is handled as a "TREAT as WITHDRAW" [RFC7606].

4.1. Validation of FS NLRI (FSv1 or FSv2)

Flow specifications received from a BGP peer that are accepted in the respective Adj-RIB-In are used as input to the route selection process. Although the forwarding attributes of the two routes for the same prefix may be the same, BGP is still required to perform its path selection algorithm in order to select the correct set of attributes to advertise.

The first step of the BGP Route selection procedure (section 9.1.2 of [RFC4271]) is to exclude from the selection procedure routes that are considered unfeasible. In the context of IP routing information, this is used to validate that the NEXT_HOP Attribute of a given route is resolvable.

The concept can be extended in the case of the Flow Specification NLRI to allow other validation procedures.

The FSv2 validation process validates the FSv2 NLRI with following unicast routes received over the same AFI (1 or 2) but different SAFIs:

* Flow specification routes (FSv1 or FSv2) received over SAFI=133 will be validated against SAFI=1,

* Flow Specification routes (FSv1 or FSv2) received over SAFI=134 will be validated against SAFI=128, and

* Flow Specification routes (FSv1 or FSv2) [AFI =1, 2] received over SAFI=77 will be validated using only the Outer Flow Spec against SAFI = 133.
The FSv2 validates L2 FSv2 NLRI with the following L2 routes received over the same AFI (25), but a different SAFI:

* Flow specification routes (FSv1 or FSv2) received over SAFI=135 are validated against SAFI=128.

In the absence of explicit configuration, a Flow specification NLRI (FSv1 or FSv2) MUST be validated such that it is considered feasible if and only if all of the conditions are true:

a) A destination prefix component is embedded in the Flow Specification,

b) One of the following conditions holds true:

- 1. The originator of the Flow Specification matches the originator of the best-match unicast route for the destination prefix embedded in the flow specification (this is the unicast route with the longest possible prefix length covering the destination prefix embedded in the flow specification).

- 2. The AS_PATH attribute of the flow specification is empty or contains only an AS_CONFED_SEQUENCE segment [RFC5065].
  
  o 2a. This condition should be enabled by default.
  
  o 2b. This condition may be disabled by explicit configuration on a BGP Speaker,
  
  o 2c. As an extension to this rule, a given non-empty AS_PATH (besides AS_CONFED_SEQUENCE segments) MAY be permitted by policy.

  c) There are no "more-specific" unicast routes when compared with the flow destination prefix that have been received from a different neighbor AS than the best-match unicast route, which has been determined in rule b.

However, part of rule a may be relaxed by explicit configuration, permitting Flow Specifications that include no destination prefix component. If such is the case, rules b and c are moot and MUST be disregarded.

By "originator" of a BGP route, we mean either the address of the originator in the ORIGINATOR_ID Attribute [RFC4456] or the source address of the BGP peer, if this path attribute is not present.
A BGP implementation MUST enforce that the AS in the left-most position of the AS_PATH attribute of a Flow Specification Route (FSv1 or FSv2) received via the Exterior Border Gateway Protocol (eBGP) matches the AS in the left-most position of the AS_PATH attribute of the best-match unicast route for the destination prefix embedded in the Flow Specification (FSv1 or FSv2) NLRI.

The best-match unicast route may change over time independently of the Flow Specification NLRI (FSv1 or FSv2). Therefore, a revalidation of the Flow Specification MUST be performed whenever unicast routes change. Revalidation is defined as retesting rules a to c as described above.

4.2. Validation of Flow Specification Actions

Flow Specifications may be mapped to actions using Extended Communities or a Wide Communities. The FSv2 actions in Extended Communities and Wide communities can be associated with large number of NLRIs.

The ordering of precedence for these actions in the case when the user-defined order is the same follows the precedence of the FSv2 NLRI action TLV values (lowest to highest). User-defined order is the same when the order value for action is the same. All Extended Community actions MUST be translated to the user-defined order data format for internal comparison. By default, all Extended Community actions SHOULD be translated to a single value.

Actions may conflict, duplicate, or complement other actions. An example of conflict is the packet rate limiting by byte and by packet. An example of a duplicate is the request to copy or sample a packet under one of the redirect functions (RDIPv4, RDIPv6, RDIID, )

Each FSv2 actions in this document defines the potential conflicts or duplications. Specifications for new FSv2 actions outside of this specification MUST specify interactions or conflicts with any FSv2 actions (that appear in this specification or subsequent specifications).

Well-formed syntactically correct actions should be linked to a filtering rule in the order the actions should be taken. If one action in the ordered list fails, the default procedure is for the action process for this rule to stop and flag the error via system management. By explicit configuration, the action processing may continue after errors.

Implementations MAY wish to log the actions taken by FS actions (FSv1 or FSv2).
4.3. Error handling and Validation

The following two error handling rules must be followed by all BGP speakers which support FSv2:

* FSv2 NLRI having TLVs which do not have the correct lengths or syntax must be considered MALFORMED.

* FSv2 NLRIs having TLVs which do not follow the above ordering rules described in section 4.1 MUST be considered as malformed by a BGP FSv2 propagator.

The above two rules prevent any ambiguity that arises from the multiple copies of the same NLRI from multiple BGP FSv2 propagators.

A BGP implementation SHOULD treat such malformed NLRIs as ‘Treat-as-withdraw’ [RFC7606]

An implementation for a BGP speaker supporting both FSv1 and FSv2 MUST support the error handling for both FSv1 and FSv2.

5. Ordering for Flow Specification v2 (FSv2)

Flow Specification v2 allows the user to order flow specification rules and the actions associated with a rule. Each FSv2 rule has one or more match conditions and one or more actions associated with that match condition.

This section describes how to order FSv2 filters received from a peer prior to transmission to another peer. The same ordering should be used for the ordering of forwarding filtering installed based on only FSv2 filters.

Section 7.0 describes how a BGP peer that supports FSv1 and FSv2 should order the flow specification filters during the installation of these flow specification filters into FIBs or firewall engines in routers.

The BGP distribution of FSv1 NLRI and FSv2 NLRI and their associated path attributes for actions (Wide Communities and Extended Communities) is "ships-in-the-night" forwarding of different AFI/SAFI information. This recommended ordering provides for deterministic ordering of filters sent by the BGP distribution.

5.1. Ordering of FSv2 NLRI Filters

The basic principles regarding ordering of rules are simple:
1) Rule-0 (zero) is defined to be 0/0 with the "permit-all" action with BGP peers which do not support flow specification permit traffic for routes received. Rule-0 is defined to be "permit-all" for 0/0 which is the normal case for filtering for routes received by BGP.

By configuration option, the "permit-all" may be set to "deny-all" if traffic rules on routers used as BGP must have a "route" AND a firewall filter to allow traffic flow.

2) FSv2 rules are ordered based on the user-defined order numbers specified in the FSv2 NLRI (rules 1-n).

3) If multiple FSv2 NLRI have the same user-defined order, then the filters are ordered by type of FSv2 NRLI filters (see Table 1, section 4) with lowest numerical number have the best precedence.

For the same user-defined order and the same value for the FSv2 filters type, then the filters are ordered by FSv2 the component type for that FSv2 filter type (see Tables 3-6) with the lowest number having the best precedence.

For the same user-defined order, the same value of FSv2 Filter Type, and the same value for the component type, then the filters are ordered by value within the component type. Each component type defines value ordering.

For component types inherited from the FSv1 component types, there are the following two types of comparisons:

- FSv1 component value comparison for the IP prefix values, compares the length of the two prefixes. If the length is different, the longer prefix has precedence. If the length is the same, the lower IP number has precedence.

- For all other FSv1 component types, unless specified, the component data is compared using the memcmp() function defined by [ISO_IEC_9899]. For strings with the same length, the lowest string memcmp() value has precedence. For strings of different lengths, the common prefix is compared. If the common string prefix is not equal, then the string with the lowest string prefix has higher precedence. If the common prefix is equal, the longest string is considered to have higher precedence.

Notes:
* Since the user can define rules that re-order these value comparisons, this order is arbitrary and set to provide a deterministic default.

5.2. Ordering of the Actions

The FSv2 specification allows for actions to be associated by:

a) a Wide Community path attribute, or

b) an Extended Community path attribute.

Actions may be ordered by user-defined action order number from 1-n (where n is 2**16-2 and the value 2**16-1 is reserved.

By default, extended community actions are associated with default order number 32768 [0x8000] or a specific configured value for the FSv2 domain.

Action user-order number zero is defined to have an Action type of "Set Action Chain operation" (ACO) (value 0x01) that defines the default action chain process. For details on "set action chain operation" see section 3.2.1 or section 5.2.1 below.

If the user-defined action number for two actions are the same, then the actions are ordered by FSv2 action types (see Table 3 for a list of action types). If the user-defined action number and the FSv2 action types are the same, then the order must be defined by the FSv2 action.

5.2.1. Action Chain Operation (ACO)

The "Action Chain Operation" (ACO) changes the way the actions after the current action in an action chain are handled after a failure. If no action chain operations are set, then the default action of "stop upon failure" (value 0x00) will be used for the chain.

5.2.1.1. Example 1 - Default ACO

Use Case 1: Rate limit to 600 packets per second

Description: The provider will support 600 packets per second. All packets sampled for reporting purposes and packet streams over 600 packets per second will be dropped.

Suppose BGP Peer A has a
* a Wide Community action with user-defined order 10 with Traffic Sampling

* a Wide Community action with user-defined order 11 from AS 2020 that limits packet-based rate limit of 600 packets per second.

* an Extended Community from AS 2020 that does limits packet-based rate limit of 50 packets per second.

The FSv2 data base would store the following action chain:

* at user-defined action order 10
  - A user action of type 7 (traffic action) with values of Sampling and logging.

* at user-defined action order 11
  - a user action type of 12 (packet-based rate limit) with values of AS 2020 and float value for 600 packets per second (pps)

* at user-defined action order 32768 (0x8000) with type 12 and values of A user action of type 12 with values of AS 2020 and float value of 50 packets/second.

Normal action:

The match on the traffic would cause a sample of the traffic (probably with packet rate saved in logging) followed by a rate limit to 600 pps. The Extended community action would further limit the rate to 50 packets per second.

When does the action chain stop?

The default process for the action chain is to stop on failure. If there is no failure, then all three actions would occur. This is probably not what the user wants.

If there is failure at action 10 (sample and log), then there would be no rate limiting per packet (actions 11 and action 32768).

If there is failure at action 11 (rate limit to packet 600), then there would be no rate limiting per packet (action 32768).

The different options for Action chain ordering (ACO) have been worked on with NETCONF/RESTCONF configuration and actions.
5.2.1.2. Example 2: Redirect traffic over limit to processing via SFC

Use case 2: Redirect traffic over limit to processing via SFC.

Description: The normal function is for traffic over the limit to be forwarded for offline processing and reporting to a customer.

Suppose we have the following 4 actions defined for a match:

* Sent Redirect to indirection ID (0x01) with user-defined match 2 attached in wide community,

* Traffic rate limit by bytes (0x07) with user-defined match 1 attached in wide community,

* Traffic sample (0x07) sent in extended community, and

* SF classifier Info (0x0E) sent in extended community.

These 4 filters rate limit a potential DDoS attack by: a) redirect the packet to indirection ID (for slower speed processing), sample to local hardware, and forward the attack traffic via a SFC to a data collection box.

The FSv2 action list for the match would look like this

Action 0: Operation of action chain (0x01) (stop upon failure)

Action 1: Traffic Rate limit by byte (0x07)

Action 2: Redirect to Redirection ID (0x0F)

Action 32768 (0x8000) Traffic Action (0x07) Sample

Action 32768 (0x8000) SFC Classifier: (0xE)

If the redirect to a redirection ID fails, then Traffic Sample and sending the data to an SFC classifier for forwarding via SFC will not happen. The traffic is limited, but not redirect away from the network and a sample sent to DDOS processing via a SFC classifier.

Suppose the following 5 actions were defined for a FSv2 filter:

* Set Action Chain Operation (ACO) (0x01) to continue on failure (0x01) at user-order 2 attached in wide community,

* redirect to indirection ID (0x0F) at user-order 2 attached in wide community,
* traffic rate limit by bytes (0x07) with user-order 1 attached in wide community,

* Traffic sample (0x07) attached via extended community, and

* SFC classifier Info (0x0E) attached in extended community.

The FSv2 action list for the match would look like this:

Action 00: Operation of action chain (0x01) (stop upon failure)
Action 01: Traffic Rate limit by byte (0x07)
Action 02: Set Action Chain Operation (ACO) (0x01) (continue on failure)
Action 02: Redirect to Redirection ID (0F)
Action 32768 (0x8000): Traffic Action (0x07) Sample
Action 32768 (0x8000): SFC classifier (0x0E) forward via SFC [to DDoS classifier]

If the redirect to a redirection ID fails, the action chain will continue on to sample the data and enact SFC classifier actions.

5.2.2. Summary of FSv2 ordering

Operators should use user-defined ordering to clearly specify the actions desired upon a match. The FSv2 actions default ordering is specified to provide deterministic order for actions which have the same user-defined order and same type.
### FS Action

<table>
<thead>
<tr>
<th>FS Action</th>
<th>Value Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>(lowest value to highest)</td>
<td>(lowest to highest)</td>
</tr>
<tr>
<td>=================================</td>
<td>------------------------------</td>
</tr>
<tr>
<td>0x01: ACO: Action chain operation</td>
<td>Failure flag</td>
</tr>
<tr>
<td>0x02: TAIS: Traffic actions per</td>
<td>AS, then Group-ID, then</td>
</tr>
<tr>
<td>Interface group</td>
<td>Action ID</td>
</tr>
<tr>
<td>0x03-0x05 to be assigned</td>
<td>TBD</td>
</tr>
<tr>
<td>0x06: TRB: Traffic rate limit</td>
<td>AS, then float value</td>
</tr>
<tr>
<td>by bytes</td>
<td></td>
</tr>
<tr>
<td>0x07: TA: Traffic Action</td>
<td>traffic action value</td>
</tr>
<tr>
<td>0x08: RDIP: Redirect to IP</td>
<td>AS, then IP Address, then ID</td>
</tr>
<tr>
<td>0x09: TM: Traffic Marking</td>
<td>DSCP value (lowest to highest)</td>
</tr>
<tr>
<td>0x0A: AL2: Associated L2 Info.</td>
<td>TBD</td>
</tr>
<tr>
<td>0x0B: AET: Associated E-tree Info.</td>
<td>TBD</td>
</tr>
<tr>
<td>0x0C: TRP: Traffic Rate limit</td>
<td>AS, then float value</td>
</tr>
<tr>
<td>by bytes</td>
<td></td>
</tr>
<tr>
<td>0x0D: RDIPv6: Traffic Redirect to IPv6</td>
<td>AS, IPv6 value, then local Admin</td>
</tr>
<tr>
<td>0x0E: TISFC: Traffic insertion to SFC</td>
<td>SPI, then SI, the SFT</td>
</tr>
<tr>
<td>0x0F: Redirect to Indirection-ID</td>
<td>ID-type, then Generalized-ID</td>
</tr>
<tr>
<td>0x10: MPLSLA: MPLS Label stack</td>
<td>order, action, label, Exp</td>
</tr>
<tr>
<td>0x16 VLAN action</td>
<td>rewrite-actions, VALN1, VLAN2, PCP-DE1, PCP-DE2</td>
</tr>
<tr>
<td>0x17 TPID action</td>
<td>rewrite actions, TP-ID-1, TP-ID-2</td>
</tr>
</tbody>
</table>

**Figure 6-1**

6. Ordering of FS filters for BGP Peers support FSv1 and FSv2

FSv2 allows the user to order flow specification rules and the actions associated with a rule. Each FSv2 rule has one or more match conditions and one or more actions associated with each rule.

FSv1 and FSv2 filters are sent as different AFI/SAFI pairs so FSv1 and FSv2 operate as ships-in-the-night. Some BGP peers in an AS may support both FSv1 and FSv2. Other BGP peers may support FSv1 or FSv2. Some BGP will not support FSv1 or FSv2. A coherent flow specification technology must have consistent best practices for ordering the FSv1 and FSv2 filter rules.

One simple rule captures the best practice: Order the FSv1 filters after the FSv2 filter by placing the FSv1 filters after the FSv2 filters.
To operationally make this work, all flow specification filters should be included the same data base with the FSv1 filters being assigned a user-defined order beyond the normal size of FSv2 user-ordered values. A few examples, may help to illustrate this best practice.

Example 1: User ordered numbering - Suppose you might have 1,000 rules for the FSv2 filters. Assign all the FSv1 user defined rules to 1,001 (or better yet 2,000). The FSv1 rules will be ordered by the components and component values.

Example 2: Storage of actions - All FSv1 actions are defined ordered actions in FSv2. Translate your FSv1 actions into FSv2 ordered actions for storing in a common FSv1-FSv2 flow specification data base.

Example 3: Mixed Flow Specification Support -

Suppose an FSv2 peer (BGP Peer A) has the capability to send either FSv1 or FSv2. BGP Peer A peers with BGP Peers B, C, D and E.

BGP Peer B can only send FSv1 routes (NLRI + Extended Community).
BGP Peer C can send FSv2 routes (NLRI + path attributes (wide community or extended community or none)). BGP Peer D cannot send any FS routes. BGP E can send FSv2 and FSv1 routes

BGP Peer A sends FSv1 routes in its databases to BGP B. Since the FSv2 NLRI cannot be sent to the FSv1 peer, only the FSv1 NLRI is sent. BGP Peer A sends to BGP C the FSv2 routes in its database (configured or received).

BGP Peer A would not send the FSv1 NLRI or FSv2 NLRI to BGP Peer D. The BGP Peer D does not support for these NLRI.

BGP Peer A sends the NLRI for both FSv1 and FSv2 to BGP Peer E.
7. Scalability and Aspirations for FSv2

Operational issues drive the deployment of BGP flow specification as a quick and scalable way to distribute filters. The early operations accepted the fact validation of the distribution of filter needed to be done outside of the BGP distribution mechanism. Other mechanisms (NETCONF/RESTCONF or PCEP) have reply-request protocols.

These features within BGP have not changed. BGP still does not have an action-reply feature.

NETCONF/RESTCONF latest enhancements provide action/response features which scale. The combination of a quick distribution of filters via BGP and a long-term action in NETCONF/RESTCONF that ask for reporting of the installation of FSv2 filters may provide the best scalability.

The combination of NETCONF/RESTCONF network management protocols and BGP focuses each protocol on the strengths of scalability.

FSv2 will be deployed in webs of BGP peers which have some BGP peers passing FSv1, some BGP peers passing FSv2, some BGP peers passing FSv1 and FSv2, and some BGP peers not passing any routes.
The TLV encoding and deterministic behaviors of FSv2 will not deprecate the need for careful design of the distribution of flow specification filters in this mixed environment. The needs of networks for flow specification are different depending on the network topology and the deployment technology for BGP peers sending flow specification.

Suppose we have a centralized RR connected to DDoS processing sending out flow specification to a second tier of RR who distribute the information to targeted nodes. This type of distribution has one set of needs for FSv2 and the transition from FSv1 to FSv2.

Suppose we have Data Center with a 3-tier backbone trying to distribute DDoS or other filters from the spine to combinational nodes, to the leaf BGP nodes. The BGP peers may use RR or normal BGP distribution. This deployment has another set of needs for FSv2 and the transition from FSv1 to FSv2.

Suppose we have a corporate network with a few AS sending DDoS filters using basic BGP from a variety of sites. Perhaps the corporate network will be satisfied with FSv1 for a long time.

These examples are given to indicate that BGP FSv2, like so many BGP protocols, needs to be carefully tuned to aid the mitigation services within the network. This protocol suite starts the migration toward better tools using FSv2, but it does not end it. With FSv2 TLVs and deterministic actions, new operational mechanisms can start to be understood and utilized.

This FSv2 specification is merely the start of a revolution of work - not the end.

8. Optional Security Additions

This section discusses the optional BGP Security additions for BGP-FS v2 relating to BGPSEC [RFC8205] and ROA [RFC6482].

8.1. BGP FSv2 and BGPSEC

Flow specification v1 ([RFC8955] and [RFC8956]) do not comment on how BGP Flow specifications to be passed BGPSEC [RFC8205] BGP Flow Specification v2 can be passed in BGPSEC, but it is not required.

FSv1 and FSv2 may be sent via BGPSEC.
8.2. BGP FSv2 with ROA

BGP FSv2 can utilize ROAs in the validation. If BGP FSv2 is used with BGPSEC and ROA, the first thing is to validate the route within BGPSEC and second to utilize BGP ROA to validate the route origin.

The BGP-FS peers using both ROA and BGP-FS validation determine that a BGP Flow specification is valid if and only if one of the following cases:

* If the BGP Flow Specification NLRI has a IPv4 or IPv6 address in destination address match filter and the following is true:
  - A BGP ROA has been received to validate the originator, and
  - The route is the best-match unicast route for the destination prefix embedded in the match filter; or

* If a BGP ROA has not been received that matches the IPv4 or IPv6 destination address in the destination filter, the match filter must abide by the [RFC8955] and [RFC8956] validation rules as follows:
  - The originator match of the flow specification matches the originator of the best-match unicast route for the destination prefix filter embedded in the flow specification", and
  - No more specific unicast routes exist when compared with the flow destination prefix that have been received from a different neighboring AS than the best-match unicast route, which has been determined in step A.

The best match is defined to be the longest-match NLRI with the highest preference.

9. IANA Considerations

This section complies with [RFC7153].

9.1. Flow Specification V2 SAFIs

IANA is requested to assign two SAFI Values in the registry at https://www.iana.org/assignments/safi-namespace from the Standard Action Range as follows:
### 9.2. BGP Capability Code

IANA is requested to assign a Capability Code from the registry at https://www.iana.org/assignments/capability-codes/ from the IETF Review range as follows:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
<th>Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD3</td>
<td>Flow Specification V2</td>
<td>[this document]</td>
<td>IETF</td>
</tr>
</tbody>
</table>

### 9.3. Filter IP Component types

IANA is requested to indicate [this draft] as a reference on the following assignments in the Flow Specification Component Types Registry:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Destination filter</td>
<td>[RFC8955][RFC8956][this document]</td>
</tr>
<tr>
<td>2</td>
<td>Source Prefix</td>
<td>[RFC8955][RFC8956][this document]</td>
</tr>
<tr>
<td>3</td>
<td>IP Protocol</td>
<td>[RFC8955][RFC8956][this document]</td>
</tr>
<tr>
<td>4</td>
<td>Port</td>
<td>[RFC8955][RFC8956][this document]</td>
</tr>
<tr>
<td>5</td>
<td>Destination Port</td>
<td>[RFC8955][RFC8956][this document]</td>
</tr>
<tr>
<td>6</td>
<td>Source Port</td>
<td>[RFC8955][RFC8956][this document]</td>
</tr>
<tr>
<td>7</td>
<td>ICMP Type [v4 or v6]</td>
<td>[RFC8955][RFC8956][this document]</td>
</tr>
<tr>
<td>8</td>
<td>ICMP Code [v4 or v6]</td>
<td>[RFC8955][RFC8956][this document]</td>
</tr>
<tr>
<td>9</td>
<td>TCP Flags [v4]</td>
<td>[RFC8955][RFC8956][this document]</td>
</tr>
<tr>
<td>10</td>
<td>Packet Length</td>
<td>[RFC8955][RFC8956][this document]</td>
</tr>
<tr>
<td>11</td>
<td>DSCP marking</td>
<td>[RFC8955][RFC8956][this document]</td>
</tr>
<tr>
<td>12</td>
<td>Fragment</td>
<td>[RFC8955][RFC8956][this document]</td>
</tr>
<tr>
<td>13</td>
<td>Flow Label</td>
<td>[RFC8956][this document]</td>
</tr>
<tr>
<td>14</td>
<td>TTL</td>
<td>[this document]</td>
</tr>
<tr>
<td>15</td>
<td>Partial SID</td>
<td>[draft-ietf-idr-flowspec-srv6]</td>
</tr>
<tr>
<td>16</td>
<td>MPLS Label Match 1</td>
<td>[this document]</td>
</tr>
<tr>
<td>17</td>
<td>MPLS Label Match 2</td>
<td>[this document]</td>
</tr>
</tbody>
</table>
9.4. FSV2 NLRI TLV Types

IANA is requested to create the following two new registries on a new "Flow Specification v2 TLV Types" web page.

Name: BGP FSv2 TLV types
Reference: [this document]
Registration Procedures: 0x01-0x3FFF Standards Action.

<table>
<thead>
<tr>
<th>Type</th>
<th>Use</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Reserved</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x01</td>
<td>IP traffic rules</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x02</td>
<td>FSv2 Actions</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x03</td>
<td>L2 traffic rules</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x04</td>
<td>tunnel traffic rules</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x05</td>
<td>SFC AFI filter rules</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x06</td>
<td>BGP/MPLS VPN IP traffic rules</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x07</td>
<td>BGP/MPLS VPN L2 traffic rules</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x08-0x3FFF</td>
<td>Unassigned</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x4000-0x7FFF</td>
<td>Vendor specific</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x8000-0xFFFF</td>
<td>Reserved specific</td>
<td>[this document]</td>
</tr>
</tbody>
</table>
Name: BGP FSv2 Action types
Reference: [this document]
Registration Procedure: 0x01-0x3FFF Standards Action.

<table>
<thead>
<tr>
<th>Type</th>
<th>Use</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Reserved</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x01</td>
<td>ACO: Action Chain Operation</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x02</td>
<td>TAIS: Traffic actions per interface group</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x03</td>
<td>Unassigned</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x04</td>
<td>Unassigned</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x05</td>
<td>Unassigned</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x06</td>
<td>TRB: traffic rate limited by bytes</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x07</td>
<td>TA: Traffic action (terminal/sample)</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x08</td>
<td>RDIPv4: redirect IPv4</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x09</td>
<td>TM: traffic marking (DSCP)</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x0A</td>
<td>AL2: associate L2 Information</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x0B</td>
<td>AET: associate E-Tree information</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x0C</td>
<td>TRP: traffic rate limited by packets</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x0D</td>
<td>RDIPv6: Redirect to IPv6</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x0E</td>
<td>TISFC: Traffic insertion to SFC</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x0F</td>
<td>RDIID: Redirect to indirection-id</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x10</td>
<td>MPLS Label Action</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x11</td>
<td>unassigned</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x12</td>
<td>unassigned</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x13</td>
<td>unassigned</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x14</td>
<td>unassigned</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x15</td>
<td>unassigned</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x16</td>
<td>VLAN action</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x17</td>
<td>TIPD action</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x18-</td>
<td>Unassigned</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x3ff</td>
<td>Unassigned</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x4000-</td>
<td>Unassigned</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x7fff</td>
<td>Vendor assigned</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x8000-</td>
<td>Unassigned</td>
<td>[this document]</td>
</tr>
<tr>
<td>0xFFFF</td>
<td>Reserved</td>
<td>[this document]</td>
</tr>
</tbody>
</table>

9.5. Wide Community Assignments

IANA is requested to assign values in the BGP Community Container Atom Type Registry
IANA is requested to assign values from the Registered Type 1 BGP Wide Community Types:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSV2 Actions</td>
<td>TBD4</td>
<td></td>
</tr>
</tbody>
</table>

10. Security Considerations

The use of ROA improves on [RFC8955] by checking to see if the route origination. This check can improve the validation sequence for a multiple-AS environment.

The use of BGPSEC [RFC8205] to secure the packet can increase security of BGP flow specification information sent in the packet.

The use of the reduced validation within an AS [RFC9117] can provide adequate validation for distribution of flow specification within a single autonomous system for prevention of DDoS.

Distribution of flow filters may provide insight into traffic being sent within an AS, but this information should be composite information that does not reveal the traffic patterns of individuals.

11. References

11.1. Normative References

[I-D.ietf-idr-bgp-flowspec-label]


Hares, et al. Expires 8 August 2022 [Page 65]
[I-D.ietf-idr-wide-bgp-communities]
Raszuk, R., Haas, J., Lange, A., Decraene, B., Amante, S.,
and P. Jakma, "BGP Community Container Attribute", Work in
Progress, Internet-Draft, draft-ietf-idr-wide-bgp-
communities-06, 10 January 2022,
<https://www.ietf.org/archive/id/draft-ietf-idr-wide-bgp-
communities-06.txt>.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791,
DOI 10.17487/RFC7911, September 1981,

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y.,
Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack
Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001,

Border Gateway Protocol 4 (BGP-4)", RFC 4271,
DOI 10.17487/RFC4271, January 2006,

Communities Attribute", RFC 4360, DOI 10.17487/RFC4360,

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter,
"Multiprotocol Extensions for BGP-4", RFC 4760,
DOI 10.17487/RFC4760, January 2007,

System Confederations for BGP", RFC 5065,
DOI 10.17487/RFC5065, August 2007,

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route
Origin Authorizations (ROAs)", RFC 6482,
DOI 10.17487/RFC6482, February 2012,

11.2. Informative References


Authors’ Addresses

Susan Hares
Hickory Hill Consulting
7453 Hickory Hill
Saline, MI 48176
United States of America

Phone: +1-734-604-0332
Email: shares@ndzh.com

Donald Eastlake
Futurewei Technologies
2386 Panoramic Circle
Apopka, FL 32703
United States of America

Phone: +1-508-333-2270
Email: d3e3e3@gmail.com

Chaitanya Yadlapalli
ATT
United States of America

Email: cy098d@att.com

Sven Maduschke
Verizon
Germany

Email: sven.maduschke@de.verizon.com
Dissemination of Flow Specification Rules

draft-hares-idr-rfc5575bis-01.txt

Abstract

This document updates RFC5575 which defines a Border Gateway Protocol Network Layer Reachability Information (BGP NLRI) encoding format that can be used to distribute traffic flow specifications. This allows the routing system to propagate information regarding more specific components of the traffic aggregate defined by an IP destination prefix (IPv4, IPv6), MPLS addresses, L2VPN addresses, and NV03 encapsulation of IP addresses. The information is carried via the BGP, thereby reusing protocol algorithms, operational experience, and administrative processes such as inter-provider peering agreements.

There are three applications of that encoding format: 1) automation of inter-domain coordination of traffic filtering, such as what is required in order to mitigate (distributed) denial-of-service attacks; 2) enable traffic filtering in the context of a BGP/MPLS VPN service, and 3) aid centralized control of traffic in a SDN or NFV context. Some of deployments of these three applications can be handled by the strict ordering of the BGP NLRI traffic flow filters, and the strict actions encoded in the Extended Community Flow Specification actions. Other deployments (especially SDN/NFV) need to be able to allow the user to order the flow specification. Another BGP Flow Specification (version 2) is being defined for user-ordered filters, and user-ordered actions encoded in Wide Communities.

This document provides the definition of a BGP NLRI which carries traffic flow specification filters, and Extended Community values which encode the actions a routing system can take if a packet matches the traffic flow filters. The specification requires that the BGP Flow Specification traffic filters follows a string ordering, and that the BGP Flow Specification Extended Communities actions are processed in a defined order. This BGP Flow Specification is denoted as BGP Flow Specification version 1.
There are three applications of that encoding format: 1) automation of inter-domain coordination of traffic filtering, such as what is required in order to mitigate (distributed) denial-of-service attacks; 2) enable traffic filtering in the context of a BGP/MPLS VPN service, and 3) aid centralized control of traffic in a SDN or NFV context. Some of deployments of these three applications can be handled by the strict ordering of the BGP NLRI traffic flow filters, and the strict actions encoded in the Extended Community Flow Specification actions. Other deployments (especially SDN/NFV) need to be able to allow the user to order the flow specification. Another BGP Flow Specification (version 2) is being defined for user-ordered filters, and user-ordered actions encoded in Wide Communities.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 9, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

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

1. Introduction ......................................... 4
2. Definitions of Terms Used in This Memo .............. 6
3. Flow Specifications ................................. 6
4. Traffic Filtering ............................... 7
   4.1. Support for other AFIs ......................... 8
5. Dissemination of IPv4 Flow Specification Information .. 8
   5.1. Length Encoding .............................. 9
   5.2. NLRI Value Encoding .......................... 9
      5.2.1. Type 1 - Destination Prefix ............... 13
      5.2.2. Type 2 - Source Prefix .................... 13
      5.2.3. Type 3 - Source Prefix .................... 13
      5.2.4. Type 4 - Port .......................... 14
      5.2.5. Type 5 - Destination Port ................. 14
      5.2.6. Type 6 - Destination Port ................. 14
      5.2.7. Type 7 - ICMP type ........................ 14
      5.2.8. Type 8 - ICMP code ....................... 15
      5.2.9. Type 9 - ICMP code ....................... 15
      5.2.10. Type 10 - Packet length .................. 15
      5.2.11. Type 11 - DSCP (Diffserv Code Point) .... 16
      5.2.12. Type 12 - Fragment ...................... 16
      5.2.13. Examples of Encodings ..................... 16
   5.3. Ordering of Traffic Filtering Rules ............. 17
   5.4. Validation Procedure .......................... 19
6. Traffic Filtering Actions .......................... 20
   6.1. Traffic Rate in bytes (sub-type 0x06) .......... 21
   6.2. Traffic-action (sub-type 0x07) ................ 22
   6.3. IP Redirect (sub-type 0x08) ................... 22
   6.4. Traffic Marking (sub-type 0x09) ............... 23
   6.5. Rules on Traffic Action interference .......... 23
7. Dissemination of Traffic Filtering in BGP/MPLS VPN Networks ..... 23
   7.1. Validation Procedures for BGP/MPLS VPNs ....... 24
   7.2. Traffic Actions Rules ........................ 24
8. Limitations of Previous Traffic Filtering Efforts ....... 24
   8.1. Limitations in Previous DDOS Traffic Filtering Efforts 24
   8.2. Limitations in Previous BGP/MPLS Traffic Monitoring . 25
   8.3. Limitations in BGP Flow Specification for SDN/NFV
        Applications .................................. 26
9. Traffic Monitoring .................................. 26
10. IANA Considerations ................................ 26
   10.1. AFI/SAFI Definitions ........................ 26
   10.2. Flow Component definitions ................... 26
   10.3. Extended Community Flow Specification Actions .... 28
11. Security Considerations .......................... 28
12. Original RFC5575 authors ........................ 29
13. Acknowledgements .................................. 29
14. References .......................................... 29
1. Introduction

Modern IP routers contain both the capability to forward traffic according to IP prefixes as well as to classify, shape, rate limit, filter, or redirect packets based on administratively defined policies.

These traffic policy mechanisms allow the router to define match rules that operate on multiple fields of the packet header. Actions such as the ones described above can be associated with each rule.

The n-tuple consisting of the matching criteria defines an aggregate traffic flow specification. The matching criteria can include elements such as source and destination address prefixes, IP protocol, and transport protocol port numbers.

This document defines a general procedure to encode flow specification rules for aggregated traffic flows so that they can be distributed as a BGP [RFC5575] NLRI. Additionally, we define the required mechanisms to utilize this definition to the problem of immediate concern to the authors: intra- and inter-provider distribution of traffic filtering rules to filter (distributed) denial-of-service (DoS) attacks.

By expanding routing information with flow specifications, the routing system can take advantage of the ACL (Access Control List) or firewall capabilities in the router’s forwarding path. Flow specifications can be seen as more specific routing entries to a unicast prefix and are expected to depend upon the existing unicast data information.

A flow specification received from an external autonomous system will need to be validated against unicast routing before being accepted. If the aggregate traffic flow defined by the unicast destination prefix is forwarded to a given BGP peer, then the local system can safely install more specific flow rules that may result in different forwarding behavior, as requested by this system.

The key technology components required to address the class of problems targeted by this document are:

1. Efficient point-to-multipoint distribution of control plane information.
2. Inter-domain capabilities and routing policy support.

3. Tight integration with unicast routing, for verification purposes.

Items 1 and 2 have already been addressed using BGP for other types of control plane information. Close integration with BGP also makes it feasible to specify a mechanism to automatically verify flow information against unicast routing. These factors are behind the choice of BGP as the carrier of flow specification information.

As with previous extensions to BGP, this specification makes it possible to add additional information to Internet routers. These are limited in terms of the maximum number of data elements they can hold as well as the number of events they are able to process in a given unit of time. The authors believe that, as with previous extensions, service providers will be careful to keep information levels below the maximum capacity of their devices.

In many deployments of BGP Flow Specification, the flow specification information has replace existing host length route advertisements.

Experience with previous BGP extensions has also shown that the maximum capacity of BGP speakers has been gradually increased according to expected loads. Taking into account Internet unicast routing as well as additional applications as they gain popularity.

From an operational perspective, the utilization of BGP as the carrier for this information allows a network service provider to reuse both internal route distribution infrastructure (e.g., route reflector or confederation design) and existing external relationships (e.g., inter-domain BGP sessions to a customer network).

While it is certainly possible to address this problem using other mechanisms, this solution has been utilized in deployments because of the substantial advantage of being an incremental addition to already deployed mechanisms.

In current deployments, the information distributed by the flow-spec extension is originated both manually as well as automatically. The latter by systems that are able to detect malicious flows. When automated systems are used, care should be taken to ensure their correctness as well as to limit the number and advertisement rate of flow routes.

This specification defines required protocol extensions to address most common applications of IPv4 unicast and VPNv4 unicast filtering.
The same mechanism can be reused and new match criteria added to address similar filtering needs for other BGP address families such as:

- IPv6 [I-D.ietf-idr-flow-spec-v6],
- MAC address for L2VPN [I-D.ietf-idr-flowspec-l2vpn],
- NV03 encapsulation [I-D.ietf-idr-flowspec-nvo3] and,
- MPLS ([I-D.ietf-idr-flowspec-mpls-match], [I-D.ietf-idr-bgp-flowspec-label]).

These additions to BGP Flow Specification IPv4 are included in a separate documents to allow implementers the choice of implementing portions of the BGP Flow specification.

2. Definitions of Terms Used in This Memo

NLRI - Network Layer Reachability Information.
RIB - Routing Information Base.
Loc-RIB - Local RIB.
AS - Autonomous System number.
VRF - Virtual Routing and Forwarding instance.
PE - Provider Edge router

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

3. Flow Specifications

A flow specification is an n-tuple consisting of several matching criteria that can be applied to IP traffic. A given IP packet is said to match the defined flow if it matches all the specified criteria.

A given flow may be associated with a set of attributes, depending on the particular application; such attributes may or may not include reachability information (i.e., NEXT_HOP). Well-known or AS-specific community attributes can be used to encode a set of predetermined actions.
A particular application is identified by a specific (Address Family Identifier, Subsequent Address Family Identifier (AFI, SAFI)) pair [RFC4760] and corresponds to a distinct set of RIBs. Those RIBs should be treated independently from each other in order to assure non-interference between distinct applications.

BGP itself treats the NLRI as an opaque key to an entry in its databases. Entries that are placed in the Loc-RIB are then associated with a given set of semantics, which is application dependent. This is consistent with existing BGP applications. For instance, IP unicast routing (AFI=1, SAFI=1) and IP multicast reverse-path information (AFI=1, SAFI=2) are handled by BGP without any particular semantics being associated with them until installed in the Loc-RIB.

Standard BGP policy mechanisms, such as UPDATE filtering by NLRI prefix and community matching, SHOULD apply to the Flow specification defined NLRI-type. Network operators can also control propagation of such routing updates by enabling or disabling the exchange of a particular (AFI, SAFI) pair on a given BGP peering session.

4. Traffic Filtering

Traffic filtering policies have been traditionally considered to be relatively static. Limitations of the static mechanisms caused this mechanism to be designed for the three new applications of traffic filtering (prevention of traffic-based, denial-of-service (DOS) attacks, traffic filtering in the context of BGP/MPLS VPN service, and centralized traffic control for SDN/NFV networks) requires coordination among service providers and/or coordination among the AS within a service provider. Section 8 has details on the limitation of previous mechanisms and why BGP Flow Specification version 1 provides a solution for to prevent DOS and aid BGP/MPLS VPN filtering rules.

This flow specification NLRI defined above to convey information about traffic filtering rules for traffic that should be discarded or handled in manner specified by a set of pre-defined actions (which are defined in BGP Extended Communities). This mechanism is primarily designed to allow an upstream autonomous system to perform inbound filtering in their ingress routers of traffic that a given downstream AS wishes to drop.

In order to achieve this goal, this draft specifies two application specific NLRI identifiers that provide traffic filters, and a set of actions encoding in BGP Extended Communities. The two application specific NLRI identifiers are:
IPv4 flow specification identifier (AFI=1, SAFI=133) along with specific semantic rules for IPv4 routes, and

BGP NLRI type (AFI=1, SAFI=134) value, which can be used to propagate traffic filtering information in a BGP/MPLS VPN environment.

Distribution of the IPv4 Flow specification is described in section 6, and distribution of BGP/MPLS traffic flow specification is described in section 8. The traffic filtering actions are described in section 7.

4.1. Support for other AFIs

Other documents shown in table 5 provide the application identifiers for IPv6, L2VPN, NV03 and MPLS. However, to provide backward compatibility with [RFC5575] documents adhering to this specification do not need to support IPv6, L2VPN, NV03, and MPLS AFI/SAFIs.

Table 5 - AFI/SAFI values vs. application

<table>
<thead>
<tr>
<th>AFI</th>
<th>SAFI</th>
<th>Application</th>
<th>Document</th>
<th>Req</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>133</td>
<td>DDOS</td>
<td>this document</td>
<td>Yes</td>
</tr>
<tr>
<td>1</td>
<td>134</td>
<td>BGP/MPLS</td>
<td>this document</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>133</td>
<td>DDOS</td>
<td>draft-ietf-idr-flow-spec-v6</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>134</td>
<td>BGP/MPLS</td>
<td>draft-ietf-idr-flow-spec-v6</td>
<td>No</td>
</tr>
<tr>
<td>25</td>
<td>133</td>
<td>DDOS</td>
<td>draft-ietf-idr-flowspec-l2vpn</td>
<td>No</td>
</tr>
<tr>
<td>25</td>
<td>134</td>
<td>BGP/MPLS</td>
<td>draft-ietf-idr-flowspec-l2vpn</td>
<td>No</td>
</tr>
<tr>
<td>TBD</td>
<td>133</td>
<td>DDOS</td>
<td>draft-ietf-idr-flowspec-mpls-label</td>
<td>No</td>
</tr>
<tr>
<td>TBD</td>
<td>134</td>
<td>BGP/MPLS</td>
<td>draft-ietf-idr-flowspec-mpls-label</td>
<td>No</td>
</tr>
<tr>
<td>TBD</td>
<td>133</td>
<td>DDOS</td>
<td>draft-ietf-idr-flowspec-nv03</td>
<td>No</td>
</tr>
<tr>
<td>TBD</td>
<td>134</td>
<td>BGP/MPLS</td>
<td>draft-ietf-idr-flowspec-nv03</td>
<td>No</td>
</tr>
</tbody>
</table>

5. Dissemination of IPv4 Flow Specification Information

We define a "Flow Specification" NLRI type that may include several components such as destination prefix, source prefix, protocol, ports, and others (see Tables 1-4 below). This NLRI is treated as an opaque bit string prefix by BGP. Each bit string identifies a key to a database entry with which a set of attributes can be associated.
This NLRI information is encoded using MP_REACH_NLRI and MP_UNREACH_NLRI attributes as defined in [RFC4760]. Whenever the corresponding application does not require Next-Hop information, this shall be encoded as a 0-octet length Next Hop in the MP_REACH_NLRI attribute and ignored on receipt.

The NLRI field of the MP_REACH_NLRI and MP_UNREACH_NLRI is encoded as a 1- or 2-octet NLRI length field followed by a variable-length NLRI value. The NLRI length is expressed in octets.

```
+------------------------------+  
|    length (0xnn or 0xfn nn)  |  
+------------------------------+  
|    NLRI value (variable)    |  
+------------------------------+  
```

Figure 1: Flow-spec NLRI for IPv4

Implementations wishing to exchange flow specification rules MUST use BGP’s Capability Advertisement facility to exchange the Multiprotocol Extension Capability Code (Code 1) as defined in [RFC4760]. The (AFI, SAFI) pair carried in the Multiprotocol Extension Capability MUST be the same as the one used to identify a particular application that uses this NLRI-type.

5.1. Length Encoding

- If the NLRI length value is smaller than 240 (0xf0 hex), the length field can be encoded as a single octet.
- Otherwise, it is encoded as an extended-length 2-octet value in which the most significant nibble of the first byte is all ones.

In figure 1 above, values less-than 240 are encoded using two hex digits (0xnn). Values above 240 are encoded using 3 hex digits (0xfnnn). The highest value that can be represented with this encoding is 4095. The value 241 is encoded as 0xf0f1.

5.2. NLRI Value Encoding

The Flow specification NLRI-type consists of several optional subcomponents. A specific packet is considered to match the flow specification when it matches the intersection (AND) of all the components present in the specification. The encoding of each of the NLRI components begins with a type field as listed in Table 1-4. Sections 4.2.1 to 4.2.12 contain the specific encodings for the IPv4 IP layer and transport layer headings. Additional filters encodings for IPv6, L2VPN MAC Addresses, MPLS labels, and encapsulations for
Data Centers (e.g. NVO3) related are described in other documents referenced above.

Flow specification components must follow strict type ordering by increasing numerical order. A given component type may or may not be present in the specification, but if present, it MUST precede any component of higher numeric type value.

If a given component type within a prefix in unknown, the prefix in question cannot be used for traffic filtering purposes by the receiver. Since a flow specification has the semantics of a logical AND of all components, if a component is FALSE, by definition it cannot be applied. However, for the purposes of BGP route propagation, this prefix should still be transmitted since BGP route distribution is independent on NLRI semantics.

The <type, value< encoding is chosen in order to allow for future extensibility.
**Table 1 - NLRI Types (IP values)**

<table>
<thead>
<tr>
<th>Type</th>
<th>NLRI component</th>
<th>Document</th>
<th>Req</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IPv4 Destination Prefix</td>
<td>this document</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>IPv6 Destination Prefix</td>
<td>draft-ietf-idr-flow-spec-v6</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>IPv4 Source Prefix</td>
<td>this document</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>IPv6 Source Prefix</td>
<td>draft-ietf-idr-flow-spec-v6</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>IPv4 Protocol</td>
<td>this document</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>IPv6 Next Header</td>
<td>draft-ietf-idr-flow-spec-v6</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Transport Port</td>
<td>this document</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>(TCP/UDP source or</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>destination port</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Destination Port</td>
<td>this document</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>(TCP or UDP)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Source Port (TCP/UDP)</td>
<td>this document</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>ICMP type</td>
<td>this document</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>ICMP Code</td>
<td>this document</td>
<td>No</td>
</tr>
<tr>
<td>9</td>
<td>TCP flags</td>
<td>this document</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>IP Packet length</td>
<td>this document</td>
<td>No</td>
</tr>
<tr>
<td>11</td>
<td>DSCP</td>
<td>this document</td>
<td>No</td>
</tr>
<tr>
<td>12</td>
<td>IPv4 Fragment</td>
<td>this document</td>
<td>No</td>
</tr>
<tr>
<td>13</td>
<td>IPv6 Flow Label</td>
<td>draft-ietf-idr-flow-spec-v6</td>
<td>No</td>
</tr>
</tbody>
</table>

**Table 2 - NLRI Types (L2VPN values)**

<table>
<thead>
<tr>
<th>Type</th>
<th>NLRI component</th>
<th>Document</th>
<th>Req</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD1</td>
<td>MPLS Label on label stack</td>
<td>draft-ietf-idr-flowspec-mpls-match</td>
<td>No</td>
</tr>
<tr>
<td>TBD2</td>
<td>MPLS EXP bits on top of label stack</td>
<td>draft-ietf-idr-flowspec-mpls-match</td>
<td>No</td>
</tr>
</tbody>
</table>
Table 3 - NLRI Types (L2VPN values)

<table>
<thead>
<tr>
<th>Type</th>
<th>NLRI component</th>
<th>Document</th>
<th>Req</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD*</td>
<td>Ethernet type</td>
<td>draft-ietf-idr-flowspec-l2vpn</td>
<td>No</td>
</tr>
<tr>
<td>14</td>
<td>Flow Label</td>
<td>draft-ietf-idr-flowspec-l2vpn</td>
<td>No</td>
</tr>
<tr>
<td>15</td>
<td>Source MAC</td>
<td>draft-ietf-idr-flowspec-l2vpn</td>
<td>No</td>
</tr>
<tr>
<td>16</td>
<td>Destination MAC</td>
<td>draft-ietf-idr-flowspec-l2vpn</td>
<td>No</td>
</tr>
<tr>
<td>17</td>
<td>DSAP in LLC</td>
<td>draft-ietf-idr-flowspec-l2vpn</td>
<td>No</td>
</tr>
<tr>
<td>18</td>
<td>SSAP in LLC</td>
<td>draft-ietf-idr-flowspec-l2vpn</td>
<td>No</td>
</tr>
<tr>
<td>19</td>
<td>LLC control field</td>
<td>draft-ietf-idr-flowspec-l2vpn</td>
<td>No</td>
</tr>
<tr>
<td>20</td>
<td>SNAP</td>
<td>draft-ietf-idr-flowspec-l2vpn</td>
<td>No</td>
</tr>
<tr>
<td>21</td>
<td>VLAN ID</td>
<td>draft-ietf-idr-flowspec-l2vpn</td>
<td>No</td>
</tr>
<tr>
<td>22</td>
<td>VLAN COS</td>
<td>draft-ietf-idr-flowspec-l2vpn</td>
<td>No</td>
</tr>
<tr>
<td>23</td>
<td>Inner VLAN ID</td>
<td>draft-ietf-idr-flowspec-l2vpn</td>
<td>No</td>
</tr>
<tr>
<td>24</td>
<td>Inner VLAN COS</td>
<td>draft-ietf-idr-flowspec-l2vpn</td>
<td>No</td>
</tr>
</tbody>
</table>

*conflict between IPv6 filters and L2VPN filters means this idea type must be renumbered.

Table 4 - NV03 Encapsulations

<table>
<thead>
<tr>
<th>Type</th>
<th>NLRI component</th>
<th>Document</th>
<th>Req</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD3</td>
<td>Delimiter type</td>
<td>draft-ietf-idr-flowspec-nv03</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>(VXLAN or NVGRE)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TBD4</td>
<td>VNID</td>
<td>draft-ietf-idr-flowspec-nv03</td>
<td>No</td>
</tr>
<tr>
<td>TBD5</td>
<td>Flow ID (NVGRE)</td>
<td>draft-ietf-idr-flowspec-nv03</td>
<td>No</td>
</tr>
</tbody>
</table>

*conflict between IPv6 filters and L2VPN filters means this idea type must be renumbered.
5.2.1. Type 1 - Destination Prefix

Encoding: <type (1 octet), prefix length (1 octet), prefix>

Defines: the destination prefix to match. Prefixes are encoded as in BGP UPDATE messages, a length in bits is followed by enough octets to contain the prefix information.

5.2.2. Type 2 - Source Prefix

Encoding: <type (1 octet), prefix-length (1 octet), prefix>

Defines the source prefix to match.

5.2.3. Type 3 - Source Prefix

Encoding:<type (1 octet), [op, value]+>

Contains a set of {operator, value} pairs that are used to match the IP protocol value byte in IP packets.

The operator byte is encoded as:

```
+---+---+---+---+---+---+---+---+
| e | a |  len  | 0 | lt | gt | eq |
+---+---+---+---+---+---+---+---+
```

Numerical operator

e - end-of-list bit. Set in the last {op, value} pair in the list.

a - AND bit. If unset, the previous term is logically ORed with the current one. If set, the operation is a logical AND. It should be unset in the first operator byte of a sequence. The AND operator has higher priority than OR for the purposes of evaluating logical expressions.

len - length of the value field for this operand is given as (1 << len).

lt - less than comparison between data and value.

gt - greater than comparison between data and value.

eq - equality between data and value.
The bits lt, gt, and eq can be combined to produce "less or equal", "greater or equal", and inequality values.

5.2.4. Type 4 - Port

Encoding:<type (1 octet), [op, value]+>>

Defines a list of {operation, value} pairs that matches source OR destination TCP/UDP ports. This list is encoded using the numeric operand format defined above. Values are encoded as 1- or 2-byte quantities.

Port, source port, and destination port components evaluate to FALSE if the IP protocol field of the packet has a value other than TCP or UDP, if the packet is fragmented and this is not the first fragment, or if the system in unable to locate the transport header. Different implementations may or may not be able to decode the transport header in the presence of IP options or Encapsulating Security Payload (ESP) NULL [RFC4303] encryption.

5.2.5. Type 5 - Destination Port

Encoding:<type (1 octet), [op, value]+>>

Defines a list of {operation, value} pairs used to match the destination port of a TCP or UDP packet. Values are encoded as 1- or 2-byte quantities.

5.2.6. Type 6 - Destination Port

Encoding:<type (1 octet), [op, value]+>>

Defines a list of {operation, value} pairs used to match the source port of a TCP or UDP packet. Values are encoded as 1- or 2-byte quantities.

5.2.7. Type 7 - ICMP type

Encoding:<type (1 octet), [op, value]+>>

Defines a list of {operation, value} pairs used to match the type field of an ICMP packet. Values are encoded using a single byte.

The ICMP type and code specifiers evaluate to FALSE whenever the protocol value is not ICMP.
5.2.8. Type 8 - ICMP code

Encoding:<type (1 octet), [op, value]+>

Defines a list of {operation, value} pairs used to match the code field of an ICMP packet. Values are encoded using a single byte.

5.2.9. Type 9 - ICMP code

Encoding:<type (1 octet), [op, bitmask]+>

Bitmask values can be encoded as a 1- or 2-byte bitmask. When a single byte is specified, it matches byte 13 of the TCP header [RFC0793], which contains bits 8 through 15 of the 4th 32-bit word. When a 2-byte encoding is used, it matches bytes 12 and 13 of the TCP header with the data offset field having a "don’t care" value.

As with port specifiers, this component evaluates to FALSE for packets that are not TCP packets.

This type uses the bitmask operand format, which differs from the numeric operator format in the lower nibble.

```
| e | a | len | 0 | 0 | not | m |
```

Bitmask format

- **e, a, len** - Most significant nibble: (end-of-list bit, AND bit, and length field), as defined for in the numeric operator format.
- **not** - NOT bit. If set, logical negation of operation.
- **m** - Match bit. If set, this is a bitwise match operation defined as "(data AND value) == value"; if unset, (data AND value) evaluates to TRUE if any of the bits in the value mask are set in the data

5.2.10. Type 10 - Packet length

Encoding:<type (1 octet), [op, bitmask]>

Defines match on the total IP packet length (excluding Layer 2 but including IP header). Values are encoded using 1- or 2-byte quantities.
5.2.11. Type 11 - DSCP (Diffserv Code Point)

Encoding: <type (1 octet), [op, value]>

Defines a list of {operation, value} pairs used to match the 6-bit DSCP field [RFC2474]. Values are encoded using a single byte, where the two most significant bits are zero and the six least significant bits contain the DSCP value.

5.2.12. Type 12 - Fragment

Encoding: <type (1 octet), [op, bitmask]>

Uses bitmask operand format defined above in section 5.2.9.

0 1 2 3 4 5 6 7
+---+---+---+---+---+---+---+---+
|   Reserved    |LF |FF |IsF|DF |
+---+---+---+---+---+---+---+---+

Bitmask values:

Bit 7 - Don’t fragment (DF)
Bit 6 - Is a fragment (IsF)
Bit 5 - First fragment (FF)
Bit 4 - Last fragment (LF)

5.2.13. Examples of Encodings

An example of a flow specification encoding for: "all packets to 10.0.1/24 and TCP port 25".

+------------------+-+----------+----------+
| destination      | proto    | port     |
+------------------+-+----------+----------+
| 0x01 18 0a 00 01 | 03 81 06 | 04 81 19 |
+------------------+-+----------+----------+

Decode for protocol:
An example of a flow specification encoding for: "all packets to 10.0.1/24 from 192/8 and port (range [137, 139] or 8080)".

```plaintext
+------------------+----------+-------------------------+
| destination      | source   | port                    |
+------------------+----------+-------------------------+
| 0x01 18 0a 01 01 | 02 08 c0 | 04 03 89 45 8b 91 1f 90 |
+------------------+----------+-------------------------+
```

Decode for port:

```plaintext
+--------+----------+------------------------------+
| Value  |          |                              |
+--------+----------+------------------------------+
|   0x04 | type     |                              |
|   0x03 | operator | size=1, >=                    |
|   0x89 | value    | 137                          |
|   0x45 | operator | "AND", value size=1, <=      |
|   0x8b | value    | 139                          |
|   0x91 | operator | end-of-list, value-size=2, =  |
| 0x1f90 | value    | 8080                         |
+--------+----------+------------------------------+
```

This constitutes an NLRI with an NLRI length of 16 octets.

5.3. Ordering of Traffic Filtering Rules

With traffic filtering rules, more than one rule may match a particular traffic flow. Thus, it is necessary to define the order at which rules get matched and applied to a particular traffic flow. This ordering function must be such that it must not depend on the arrival order of the flow specification’s rules and must be constant in the network.

The relative order of two flow specification rules is determined by comparing their respective components. The algorithm starts by comparing the left-most components of the rules. If the types differ, the rule with lowest numeric type value has higher precedence (and thus will match before) than the rule that doesn’t contain that
component type. If the component types are the same, then a type-specific comparison is performed.

For IP prefix values (IP destination and source prefix) precedence is given to the lowest IP value of the common prefix length; if the common prefix is equal, then the most specific prefix has precedence.

For all other component types, unless otherwise specified, the comparison is performed by comparing the component data as a binary string using the memcmp() function as defined by the ISO C standard. For strings of different lengths, the common prefix is compared. If equal, the longest string is considered to have higher precedence than the shorter one.

Pseudocode:

flow_rule_cmp (a, b)
{
    comp1 = next_component(a);
    comp2 = next_component(b);
    while (comp1 || comp2) {
        // component_type returns infinity on end-of-list
        if (component_type(comp1) < component_type(comp2)) {
            return A_HAS_PRECEDENCE;
        }
        if (component_type(comp1) > component_type(comp2)) {
            return B_HAS_PRECEDENCE;
        }

        if (component_type(comp1) == IP_DESTINATION || IP_SOURCE) {
            common = MIN(prefix_length(comp1), prefix_length(comp2));
            cmp = prefix_compare(comp1, comp2, common);
            // not equal, lowest value has precedence
            // equal, longest match has precedence
        } else {
            common =
                MIN(component_length(comp1), component_length(comp2));
            cmp = memcmp(data(comp1), data(comp2), common);
            // not equal, lowest value has precedence
            // equal, longest string has precedence
        }
    }
    return EQUAL;
}
When other AFI families are specified for BGP Flow specifications, this logic MUST be expanded. Other AFI families include IPv6, MPLS, L2VPN, and NV03 encapsulation.

5.4. Validation Procedure

Flow specifications received from a BGP peer and that are accepted in the respective Adj-RIB-In are used as input to the route selection process. Although the forwarding attributes of two routes for the same flow specification prefix may be the same, BGP is still required to perform its path selection algorithm in order to select the correct set of attributes to advertise.

The first step of the BGP Route Selection procedure (Section 9.1.2 of [RFC4271]) is to exclude from the selection procedure routes that are considered non-feasible. In the context of IP routing information, this step is used to validate that the NEXT_HOP attribute of a given route is resolvable.

The concept can be extended, in the case of flow specification NLRI, to allow other validation procedures.

A flow specification NLRI must be validated such that it is considered feasible if and only if:

a) The originator of the flow specification matches the originator of the best-match unicast route for the destination prefix embedded in the flow specification.

b) There are no more specific unicast routes, when compared with the flow destination prefix, that have been received from a different neighboring AS than the best-match unicast route, which has been determined in step a).

By originator of a BGP route, we mean either the BGP originator path attribute, as used by route reflection, or the transport address of the BGP peer, if this path attribute is not present.

The underlying concept is that the neighboring AS that advertises the best unicast route for a destination is allowed to advertise flow-spec information that conveys a more or equally specific destination prefix. Thus, as long as there are no more specific unicast routes, received from a different neighboring AS, which would be affected by that filtering rule.

The neighboring AS is the immediate destination of the traffic described by the flow specification. If it requests these flows to be dropped, that request can be honored without concern that it
represents a denial of service in itself. Supposedly, the traffic is being dropped by the downstream autonomous system, and there is no added value in carrying the traffic to it.

BGP implementations MUST also enforce that the AS_PATH attribute of a route received via the External Border Gateway Protocol (eBGP) contains the neighboring AS in the left-most position of the AS_PATH attribute. While this rule is optional in the BGP specification, it becomes necessary to enforce it for security reasons.

6. Traffic Filtering Actions

This specification defines a minimum set of filtering actions that it standardizes as BGP extended community values [RFC4360]. This is not meant to be an inclusive list of all the possible actions, but only a subset that can be interpreted consistently across the network. Additional actions can be defined as either requiring standards or as vendor specific.

Implementations SHOULD provide mechanisms that map an arbitrary BGP community value (normal or extended) to filtering actions that require different mappings in different systems in the network. For instance, providing packets with a worse-than-best-effort, per-hop behavior is a functionality that is likely to be implemented differently in different systems and for which no standard behavior is currently known. Rather than attempting to define it here, this can be accomplished by mapping a user-defined community value to platform-/network-specific behavior via user configuration.

The default action for a traffic filtering flow specification is to accept IP traffic that matches that particular rule.

This document defines the following extended communities values shown in table X in the form 0x8xnn where nn indicates the sub-type.
Table 5 - Traffic Action Extended Communities
Defined in this document

<table>
<thead>
<tr>
<th>type</th>
<th>extended community</th>
<th>encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x8006</td>
<td>traffic-rate in bytes</td>
<td>2-byte ASN, 4-byte float</td>
</tr>
<tr>
<td>0x8007</td>
<td>traffic-action</td>
<td>bitmask</td>
</tr>
<tr>
<td>0x8108</td>
<td>redirect IPv4</td>
<td>4-octet IPv4 Address, 2-octet Value</td>
</tr>
<tr>
<td>0x8208</td>
<td>redirect AS-4byte</td>
<td>4-octet AS, 2-octet Value</td>
</tr>
<tr>
<td>0x8009</td>
<td>traffic-marking</td>
<td>DSCP value</td>
</tr>
</tbody>
</table>

Encodings for these extended communities are described below.

Some traffic action communities may interfere with each other. Section x.x of this specification provides rules for handling interference between specific types of traffic actions, and error handling based on [RFC7606] in section. Each definition of a traffic action MUST specify any interface with any other traffic actions, any impact on flow specification process, and error handling per [RFC7606].

The traffic actions are processed in ascending order of the sub-type found in the BGP Extended Communities.

6.1. Traffic Rate in bytes (sub-type 0x06)

The traffic-rate extended community is a non-transitive extended community across the autonomous-system boundary and uses following extended community encoding:

The first two octets carry the 2-octet id, which can be assigned from a 2-byte AS number. When a 4-byte AS number is locally present, the 2 least significant bytes of such an AS number can be used. This value is purely informational and should not be interpreted by the implementation.

The remaining 4 octets carry the maximum rate information in IEEE floating point [IEEE.754.1985] format, units being bytes per second. A traffic-rate of 0 should result on all traffic for the particular flow to be discarded.
Interferes with: Traffic Rate in packets. Process traffic rate in bytes (sub-type 0x06) action before traffic rate action (sub-type TBD).

6.2. Traffic-action (sub-type 0x07)

The traffic-action extended community consists of 6 bytes of which only the 2 least significant bits of the 6th byte (from left to right) are currently defined.

```
+---+---+---+---+---+---+---+---+
|        reserved       | S | T |
+---+---+---+---+---+---+---+
```

where S and T are defined as:

- **T:** Terminal Action (bit 47): When this bit is set, the traffic filtering engine will apply any subsequent filtering rules (as defined by the ordering procedure). If not set, the evaluation of the traffic filter stops when this rule is applied.

- **S:** Sample (bit 46): Enables traffic sampling and logging for this flow specification.

Interferes with: No other BGP Flow Specification traffic action in this document.

6.3. IP Redirect (sub-type 0x08)

The redirect extended community allows the traffic to be redirected to a VRF routing instance that lists the specified route-target in its import policy. If several local instances match this criteria, the choice between them is a local matter (for example, the instance with the lowest Route Distinguisher value can be elected). This extended community uses the same encoding as the Route Target extended community [RFC4360].

It should be noted that the low-order nibble of the Redirect’s Type field corresponds to the Route Target Extended Community format field (Type). (See Sections 3.1, 3.2, and 4 of [RFC4360] plus Section 2 of [RFC5668].) The low-order octet (Sub-Type) of the Redirect Extended Community remains 0x08 for all three encodings of the BGP Extended Communities (AS 2-byte, AS 4-byte, and IPv4 address).

Interferes with: All other redirect functions. All redirect functions are mutually exclusive. If this redirect function exists, then no other redirect functions can be processed.
6.4. Traffic Marking (sub-type 0x09)

The traffic marking extended community instructs a system to modify the DSCP bits of a transiting IP packet to the corresponding value. This extended community is encoded as a sequence of 5 zero bytes followed by the DSCP value encoded in the 6 least significant bits of the 6th byte.

Interferes with: No other action in this document.

6.5. Rules on Traffic Action interference

The following traffic Actions may interfere with each other:

- redirect actions,
- traffic rate actions, and
- encapsulation actions.

This specification has the following rules regarding multiple traffic actions to prevent the interference:

1. All redirect actions are mutually exclusive. Presence of more than one results in no redirect.

2. If multiple rate actions are present, these are applied in ascending order of the sub-type.

3. Some actions are unique, and may operate independently. For example, an MPLS push/pop action is unique.

4. Each additional flow specification Action must specify:
   * whether it is a redirect or rate action,
   * whether the action is unique or if it interferes with other actions,
   * if the action interferes with other actions, the handling must be specified if both the action and other interfering actions exist are associated with a Flow specification NLRI.

7. Dissemination of Traffic Filtering in BGP/MPLS VPN Networks

Provider-based Layer 3 VPN networks, such as the ones using a BGP/MPLS IP VPN [RFC4364] control plane, have different traffic filtering requirements than Internet service providers. This document proposes
an additional BGP NLRI type (AFI=1, SAFI=134) value, which can be used to propagate traffic filtering information in a BGP/MPLS VPN environment.

The NLRI format for this address family consists of a fixed-length Route Distinguisher field (8 bytes) followed by a flow specification, following the encoding defined above in section x of this document. The NLRI length field shall include both the 8 bytes of the Route Distinguisher as well as the subsequent flow specification.

```
+-----------------------------+
<table>
<thead>
<tr>
<th>length (0xnn or 0xfn nn)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route Distinguisher (8 bytes)</td>
</tr>
<tr>
<td>----------------------------</td>
</tr>
<tr>
<td>NLRI value (variable)</td>
</tr>
</tbody>
</table>
+-----------------------------+
```

Figure 2: Flow-spec NLRI for MPLS

Propagation of this NLRI is controlled by matching Route Target extended communities associated with the BGP path advertisement with the VRF import policy, using the same mechanism as described in "BGP/MPLS IP VPNs" [RFC4364].

Flow specification rules received via this NLRI apply only to traffic that belongs to the VRF(s) in which it is imported. By default, traffic received from a remote PE is switched via an MPLS forwarding decision and is not subject to filtering.

Contrary to the behavior specified for the non-VPN NLRI, flow rules are accepted by default, when received from remote PE routers.

7.1. Validation Procedures for BGP/MPLS VPNs

The validation procedures are the same as for IPv4.

7.2. Traffic Actions Rules

The traffic action rules are the same as for IPv4.

8. Limitations of Previous Traffic Filtering Efforts

8.1. Limitations in Previous DDOS Traffic Filtering Efforts

The popularity of traffic-based, denial-of-service (DoS) attacks, which often requires the network operator to be able to use traffic
filters for detection and mitigation, brings with it requirements that are not fully satisfied by existing tools.

Increasingly, DoS mitigation requires coordination among several service providers in order to be able to identify traffic source(s) and because the volumes of traffic may be such that they will otherwise significantly affect the performance of the network.

Several techniques are currently used to control traffic filtering of DoS attacks. Among those, one of the most common is to inject unicast route advertisements corresponding to a destination prefix being attacked. One variant of this technique marks such route advertisements with a community that gets translated into a discard Next-Hop by the receiving router. Other variants attract traffic to a particular node that serves as a deterministic drop point.

Using unicast routing advertisements to distribute traffic filtering information has the advantage of using the existing infrastructure and inter-AS communication channels. This can allow, for instance, a service provider to accept filtering requests from customers for address space they own.

There are several drawbacks, however. An issue that is immediately apparent is the granularity of filtering control: only destination prefixes may be specified. Another area of concern is the fact that filtering information is intermingled with routing information.

The mechanism defined in this document is designed to address these limitations. We use the flow specification NLRI defined above to convey information about traffic filtering rules for traffic that should be discarded.

8.2. Limitations in Previous BGP/MPLS Traffic Monitoring

Provider-based Layer 3 VPN networks, such as the ones using a BGP/MPLS IP VPN [RFC4364] control plane, have different traffic filtering requirements than Internet service providers.

In these environments, the VPN customer network often has traffic filtering capabilities towards their external network connections (e.g., firewall facing public network connection). Less common is the presence of traffic filtering capabilities between different VPN attachment sites. In an any-to-any connectivity model, which is the default, this means that site-to-site traffic is unfiltered.

In circumstances where a security threat does get propagated inside the VPN customer network, there may not be readily available mechanisms to provide mitigation via traffic filter.
The BGP Flow Specification version 1 addresses these limitations.

8.3. Limitations in BGP Flow Specification for SDN/NFV Applications

The SDN/NFV applications which use centralized control of network traffic via dynamic distribution of traffic filters can utilize the BGP Flow Specification version 1 described in this draft with a fixed order to traffic filter matches. However, for control of large amounts of data a user-defined order to traffic matches and actions may be required.

9. Traffic Monitoring

Traffic filtering applications require monitoring and traffic statistics facilities. While this is an implementation-specific choice, implementations SHOULD provide:

- A mechanism to log the packet header of filtered traffic.
- A mechanism to count the number of matches for a given flow specification rule.

10. IANA Considerations

This section complies with [RFC7153]

10.1. AFI/SAFI Definitions

For the purpose of this work, IANA has allocated values for two SAFIs: SAFI 133 for IPv4 dissemination of flow specification rules and SAFI 134 for VPNv4 dissemination of flow specification rules.

10.2. Flow Component definitions

A flow specification consists of a sequence of flow components, which are identified by an 8-bit component type. Types must be assigned and interpreted uniquely. The current specification defines types 1 through 12, with the value 0 being reserved.

IANA created and maintains a new registry entitled: "Flow Spec Component Types". The following component types have been registered:

- Type 1 - Destination Prefix
- Type 2 - Source Prefix
- Type 3 - IP Protocol
Type 4 - Port
Type 5 - Destination port
Type 6 - Source port
Type 7 - ICMP type
Type 8 - ICMP code
Type 9 - TCP flags
Type 10 - Packet length
Type 11 - DSCP
Type 12 - Fragment

In order to manage the limited number space and accommodate several usages, the following policies defined by RFC 5226 [RFC5226] are used:

+--------------+-------------------------------+
| Range        | Policy                        |
+--------------+-------------------------------+
| 0            | Invalid value                 |
| [1 .. 12]    | Defined by this specification |
| [13 .. 127]  | Specification Required        |
| [128 .. 255] | First Come First Served       |
+--------------+-------------------------------+

The specification of a particular "flow component type" must clearly identify what the criteria used to match packets forwarded by the router is. This criteria should be meaningful across router hops and not depend on values that change hop-by-hop such as TTL or Layer 2 encapsulation.

The "traffic-action" extended community defined in this document has 46 unused bits, which can be used to convey additional meaning. IANA created and maintains a new registry entitled: "Traffic Action Fields". These values should be assigned via IETF Review rules only. The following traffic-action fields have been allocated:

47 Terminal Action
46 Sample
0-45 Unassigned
10.3. Extended Community Flow Specification Actions

The Extended Community Flow Specification Action types consists of two parts: BGP Transitive Extended Community types and a set of sub-types.

IANA has updated the following "BGP Transitive Extended Community Types" registries to contain the values listed below:

0x80 - Generic Transitive Experimental Use Extended Community Part 1 (Sub-Types are defined in the "Generic Transitive Experimental Extended Community Part 1 Sub-Types" Registry)

0x81 - Generic Transitive Experimental Use Extended Community Part 2 (Sub-Types are defined in the "Generic Transitive Experimental Extended Community Part 2 Sub-Types" Registry)

0x82 - Generic Transitive Experimental Use Extended Community Part 3 (Sub-Types are defined in the "Generic Transitive Experimental Extended Community Part 3 Sub-Types" Registry)

<table>
<thead>
<tr>
<th>RANGE</th>
<th>REGISTRATION PROCEDURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00-0xbf</td>
<td>First Come First Served</td>
</tr>
<tr>
<td>0xc0-0xff</td>
<td>IETF Review</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SUB-TYPE VALUE</th>
<th>NAME</th>
<th>REFERENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00-0x05</td>
<td>unassigned</td>
<td></td>
</tr>
<tr>
<td>0x06</td>
<td>traffic-rate</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x07</td>
<td>traffic-action</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x08</td>
<td>Flow spec redirect IPv4</td>
<td>[RFC5575] [RFC7674] [this document]</td>
</tr>
<tr>
<td>0x09</td>
<td>traffic-marking</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x10-0xff</td>
<td>Unassigned</td>
<td>[this document]</td>
</tr>
</tbody>
</table>

11. Security Considerations

Inter-provider routing is based on a web of trust. Neighboring autonomous systems are trusted to advertise valid reachability information. If this trust model is violated, a neighboring autonomous system may cause a denial-of-service attack by advertising reachability information for a given prefix for which it does not provide service.

As long as traffic filtering rules are restricted to match the corresponding unicast routing paths for the relevant prefixes, the security characteristics of this proposal are equivalent to the existing security properties of BGP unicast routing.
Where it is not the case, this would open the door to further denial-of-service attacks.

Enabling firewall-like capabilities in routers without centralized management could make certain failures harder to diagnose. For example, it is possible to allow TCP packets to pass between a pair of addresses but not ICMP packets. It is also possible to permit packets smaller than 900 or greater than 1000 bytes to pass between a pair of addresses, but not packets whose length is in the range 900-1000. Such behavior may be confusing and these capabilities should be used with care whether manually configured or coordinated through the protocol extensions described in this document.

12. Original RFC5575 authors

Barry Greene, MuPedro Marques, Jared Mauch, Danny McPherson, Robert Rasuzk, and Nischal Sheth were authors on [RFC5575], and therefore are contributing authors on this document.

Note: Any original authors that want to work on this text will be added as authors.

13. Acknowledgements

The authors would like to thank Yakov Rekhter, Dennis Ferguson, Chris Morrow, Charlie Kaufman, and David Smith for their comments for the comments on the original [RFC5575]. Chaitanya Kodeboyina helped design the flow validation procedure; and Steven Lin and Jim Washburn ironed out all the details necessary to produce a working implementation in the original [RFC5575].

Additional acknowledgements for this document will be included here.

14. References

14.1. Normative References


Hares Expires January 9, 2017 [Page 30]
14.2. Informative References

[I-D.ietf-idr-bgp-flowspec-label]

[I-D.ietf-idr-bgp-flowspec-oid]

[I-D.ietf-idr-flow-spec-v6]


Author’s Address

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI  48176
USA

Email: shares@ndzh.com
Large BGP Community
draft-heitz-idr-large-community-04

Abstract

A new type of BGP community attribute that contains communities that each hold a 4-octet AS number and a 8-octet opaque field is defined.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on March 10, 2017.
1. Introduction

A Large Community attribute is defined that encodes 12 bytes communities, suitable for 4-Octet Autonomous System Numbers that require 8 octets of locally significant opaque data.

The Large Community is specifically designed to accommodate routing policy related to 4-byte ASNs, as it allows operators to specify two 4-byte ASNs and still have room for 4 bytes for an action. For example, to make a request to AS65551 to add 3 prepends when sending a route to AS65536, one might add the Large Community 65551:303:65536. AS65551 would publish a list of large communities and their associated actions. The Large Community is opaque.

To ensure rapid and smooth adoption of the new community attribute, it must be as similar to the [RFC1997] community as possible, only bigger.
2. Large BGP Community Attribute

The Large Community Attribute is a transitive optional BGP attribute, with the Type Code (suggested 41) to be assigned by IANA. The attribute consists of a set of Large Communities. All routes with the Large Community attribute belong to the communities listed in the attribute.

Each Large Community is encoded as a 12-octet quantity, as follows:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------------------------------------------+---
| Autonomous System number                  |
+-------------------------------------------+---
| Local Data Part 1                          |
+-------------------------------------------+---
| Local Data Part 2                          |
+-------------------------------------------+---
```

Autonomous System Number: This field indicates the Autonomous System in which the Large Community has a meaning.

Local Data part 1: data set by network operator

Local Data part 2: data set by network operator

3. Textual Representation

The textual representation of the Large Community is A:B:C, where A is the Autonomous System number, B is the Local Data part 1 and C is the Local Data part 2. A ranges from 0 to 4294967295. B ranges from 0 to 4294967295. C ranges from 0 to 4294967295. A, B and C are plain decimal non-negative integers without leading zeroes. Each number must appear, even if it is 0. For example, "0:1:2" cannot be written as ":1:2". The string is expected to match the following regular expression: ^[0-9]+:[0-9]+:[0-9]+$

4. Error Handling

The error handling of Large Community is as follows:

- The Large Community attribute SHALL be considered malformed if its length is not a non-zero multiple of 12 bytes.

- An UPDATE message with a malformed Large Community attribute SHALL be handled using the approach of "treat-as-withdraw" as described in section 2 [RFC7606].
5. Security Considerations

TBD

6. Implementation status - RFC EDITOR: REMOVE BEFORE PUBLICATION

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

As of today these vendors have produced an implementation of Large BGP Community:

- Cisco IOS XR
- ExaBGP

The latest implementation news is tracked at http://largebgpcommunities.net/ [1].

7. IANA Considerations

IANA is requested to assign a BGP path attribute value for the Large Community attribute (suggested 41).

8. Acknowledgements

Thanks to Ruediger Volk, Russ White, Acee Lindem, Shyam Sethuram, Jared Mauch, Joel M. Halpern and Nick Hilliard for insightful review and comments.

9. References

9.1. Normative References

9.2. Informative References


9.3. URIs

[1] https://largebgpcommunities.net

Authors’ Addresses

Jakob Heitz
Cisco
170 West Tasman Drive
San Jose, CA  95054
USA

Email: jheitz@cisco.com

Keyur Patel
Arrcus, Inc

Email: keyur@arrcus.com

Job Snijders
NTT Communications
Theodorus Majofoeskistraat 100
Amsterdam  1065 SZ
NL

Email: job@ntt.net
Ignas Bagdonas  
Equinix  
London  
UK  
Email: ibagdona.ietf@gmail.com

Adam Simpson  
Nokia  
600 March Road  
Ottawa Ontario K2K 2E6  
Canada  
Email: adam.1.simpson@nokia.com
Dissemination of Flow Specification Rules
draft-hr-idr-rfc5575bis-03.txt

Abstract

This document updates RFC5575 which defines a Border Gateway Protocol Network Layer Reachability Information (BGP NLRI) encoding format that can be used to distribute traffic flow specifications. This allows the routing system to propagate information regarding more specific components of the traffic aggregate defined by an IP destination prefix. This draft specifies IPv4 traffic flow specifications via a BGP NLRI which carries traffic flow specification filter, and an Extended community value which encodes actions a routing system can take if the packet matches the traffic flow filters. The flow filters and the actions are processed in a fixed order. Other drafts specify IPv6, MPLS addresses, L2VPN addresses, and NV03 encapsulation of IP addresses.

This document updates RFC5575 to correct unclear specifications in the flow filters and to provide rules for actions which interfere (e.g. redirection of traffic and flow filtering).

Applications which use the bgp flow specification are: 1) application which automate of inter-domain coordination of traffic filtering, such as what is required in order to mitigate (distributed) denial-of-service attacks; 2) application which control traffic filtering in the context of a BGP/MPLS VPN service, and 3) applications with centralized control of traffic in a SDN or NFV context. Some of deployments of these three applications can be handled by the strict ordering of the BGP NLRI traffic flow filters, and the strict actions encoded in the Extended Community Flow Specification actions.
Status of This Memo

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

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

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

This Internet-Draft will expire on August 18, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction ................................................. 3
2. Definitions of Terms Used in This Memo .................. 5
3. Flow Specifications ........................................ 6
   4.1. Length Encoding ....................................... 7
   4.2. NLRI Value Encoding ................................... 7
         4.2.1. Type 1 - Destination Prefix ..................... 8
         4.2.2. Type 2 - Source Prefix .......................... 8
         4.2.3. Type 3 - IP Protocol ............................ 8
         4.2.4. Type 4 - Port .................................. 10
         4.2.5. Type 5 - Destination Port ....................... 10
         4.2.6. Type 6 - Source Port ............................ 10
         4.2.7. Type 7 - ICMP type .............................. 10
         4.2.8. Type 8 - ICMP code .............................. 11
1. Introduction

Modern IP routers contain both the capability to forward traffic according to IP prefixes as well as to classify, shape, rate limit, filter, or redirect packets based on administratively defined policies.

These traffic policy mechanisms allow the router to define match rules that operate on multiple fields of the packet header. Actions such as the ones described above can be associated with each rule.

The n-tuple consisting of the matching criteria defines an aggregate traffic flow specification. The matching criteria can include...
elements such as source and destination address prefixes, IP protocol, and transport protocol port numbers.

This document defines a general procedure to encode flow specification rules for aggregated traffic flows so that they can be distributed as a BGP [RFC5575] NLRI. Additionally, we define the required mechanisms to utilize this definition to the problem of immediate concern to the authors: intra- and inter-provider distribution of traffic filtering rules to filter (distributed) denial-of-service (DoS) attacks.

By expanding routing information with flow specifications, the routing system can take advantage of the ACL (Access Control List) or firewall capabilities in the router’s forwarding path. Flow specifications can be seen as more specific routing entries to a unicast prefix and are expected to depend upon the existing unicast data information.

A flow specification received from an external autonomous system will need to be validated against unicast routing before being accepted. If the aggregate traffic flow defined by the unicast destination prefix is forwarded to a given BGP peer, then the local system can safely install more specific flow rules that may result in different forwarding behavior, as requested by this system.

The key technology components required to address the class of problems targeted by this document are:

1. Efficient point-to-multipoint distribution of control plane information.

2. Inter-domain capabilities and routing policy support.

3. Tight integration with unicast routing, for verification purposes.

Items 1 and 2 have already been addressed using BGP for other types of control plane information. Close integration with BGP also makes it feasible to specify a mechanism to automatically verify flow information against unicast routing. These factors are behind the choice of BGP as the carrier of flow specification information.

As with previous extensions to BGP, this specification makes it possible to add additional information to Internet routers. These are limited in terms of the maximum number of data elements they can hold as well as the number of events they are able to process in a given unit of time. The authors believe that, as with previous
extensions, service providers will be careful to keep information levels below the maximum capacity of their devices.

In many deployments of BGP Flow Specification, the flow specification information has replace existing host length route advertisements.

Experience with previous BGP extensions has also shown that the maximum capacity of BGP speakers has been gradually increased according to expected loads. Taking into account Internet unicast routing as well as additional applications as they gain popularity.

From an operational perspective, the utilization of BGP as the carrier for this information allows a network service provider to reuse both internal route distribution infrastructure (e.g., route reflector or confederation design) and existing external relationships (e.g., inter-domain BGP sessions to a customer network).

While it is certainly possible to address this problem using other mechanisms, this solution has been utilized in deployments because of the substantial advantage of being an incremental addition to already deployed mechanisms.

In current deployments, the information distributed by the flow-spec extension is originated both manually as well as automatically. The latter by systems that are able to detect malicious flows. When automated systems are used, care should be taken to ensure their correctness as well as to limit the number and advertisement rate of flow routes.

This specification defines required protocol extensions to address most common applications of IPv4 unicast and VPNv4 unicast filtering. The same mechanism can be reused and new match criteria added to address similar filtering needs for other BGP address families such as IPv6 families [I-D.ietf-idr-flow-spec-v6],

2. Definitions of Terms Used in This Memo

NLRI - Network Layer Reachability Information.
RIB - Routing Information Base.
Loc-RIB - Local RIB.
AS - Autonomous System number.
VRF - Virtual Routing and Forwarding instance.
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]

3. Flow Specifications

A flow specification is an n-tuple consisting of several matching criteria that can be applied to IP traffic. A given IP packet is said to match the defined flow if it matches all the specified criteria.

A given flow may be associated with a set of attributes, depending on the particular application; such attributes may or may not include reachability information (i.e., NEXT_HOP). Well-known or AS-specific community attributes can be used to encode a set of predetermined actions.

A particular application is identified by a specific (Address Family Identifier, Subsequent Address Family Identifier (AFI, SAFI)) pair [RFC4760] and corresponds to a distinct set of RIBs. Those RIBs should be treated independently from each other in order to assure non-interference between distinct applications.

BGP itself treats the NLRI as an opaque key to an entry in its databases. Entries that are placed in the Loc-RIB are then associated with a given set of semantics, which is application dependent. This is consistent with existing BGP applications. For instance, IP unicast routing (AFI=1, SAFI=1) and IP multicast reverse-path information (AFI=1, SAFI=2) are handled by BGP without any particular semantics being associated with them until installed in the Loc-RIB.

Standard BGP policy mechanisms, such as UPDATE filtering by NLRI prefix as well as community matching and manipulation, MUST apply to the Flow specification defined NLRI-type, especially in an inter-domain environment. Network operators can also control propagation of such routing updates by enabling or disabling the exchange of a particular (AFI, SAFI) pair on a given BGP peering session.

4. Dissemination of IPv4 Flow Specification Information

We define a "Flow Specification" NLRI type (Figure 1) that may include several components such as destination prefix, source prefix, protocol, ports, and others (see Section 4.2 below). This NLRI is treated as an opaque bit string prefix by BGP. Each bit string
identifies a key to a database entry with which a set of attributes can be associated.

This NLRI information is encoded using MP_REACH_NLRI and MP_UNREACH_NLRI attributes as defined in [RFC4760]. Whenever the corresponding application does not require Next-Hop information, this shall be encoded as a 0-octet length Next Hop in the MP_REACH_NLRI attribute and ignored on receipt.

The NLRI field of the MP_REACH_NLRI and MP_UNREACH_NLRI is encoded as a 1- or 2-octet NLRI length field followed by a variable-length NLRI value. The NLRI length is expressed in octets.

```
+------------------------------+     +------------------------------+
|    length (0xnn or 0xfn nn)  |     |    length (0xnn or 0xfn nn)  |
|-------------------------------|     |-------------------------------|
|    NLRI value (variable)     |     |    NLRI value (variable)     |
+------------------------------+     |-------------------------------|
```

Figure 1: Flow-spec NLRI for IPv4

Implementations wishing to exchange flow specification rules MUST use BGP’s Capability Advertisement facility to exchange the Multiprotocol Extension Capability Code (Code 1) as defined in [RFC4760]. The (AFI, SAFI) pair carried in the Multiprotocol Extension Capability MUST be the same as the one used to identify a particular application that uses this NLRI-type.

4.1. Length Encoding

- If the NLRI length value is smaller than 240 (0xf0 hex), the length field can be encoded as a single octet.

- Otherwise, it is encoded as an extended-length 2-octet value in which the most significant nibble of the first byte is all ones.

In figure 1 above, values less-than 240 are encoded using two hex digits (0xnn). Values above 239 are encoded using 3 hex digits (0xfnnn). The highest value that can be represented with this encoding is 4095. The value 241 is encoded as 0xf0f1.

4.2. NLRI Value Encoding

The Flow specification NLRI-type consists of several optional subcomponents. A specific packet is considered to match the flow specification when it matches the intersection (AND) of all the components present in the specification.
The encoding of each of the NLRI components begins with a type field (1 octet) followed by a variable length parameter. Section 4.2.1 to Section 4.2.12 define component types and parameter encodings for the IPv4 IP layer and transport layer headers. IPv6 NLRI component types are described in [I-D.ietf-idr-flow-spec-v6].

Flow specification components must follow strict type ordering by increasing numerical order. A given component type may or may not be present in the specification, but if present, it MUST precede any component of higher numeric type value.

If a given component type within a prefix in unknown, the prefix in question cannot be used for traffic filtering purposes by the receiver. Since a flow specification has the semantics of a logical AND of all components, if a component is FALSE, by definition it cannot be applied. However, for the purposes of BGP route propagation, this prefix should still be transmitted since BGP route distribution is independent on NLRI semantics.

The <type, value> encoding is chosen in order to allow for future extensibility.

4.2.1. Type 1 - Destination Prefix

Encoding: <type (1 octet), prefix length (1 octet), prefix>

Defines: the destination prefix to match. Prefixes are encoded as in BGP UPDATE messages, a length in bits is followed by enough octets to contain the prefix information.

4.2.2. Type 2 - Source Prefix

Encoding: <type (1 octet), prefix-length (1 octet), prefix>

Defines the source prefix to match.

4.2.3. Type 3 - IP Protocol

Encoding:<type (1 octet), [op, value]+>

Contains a set of {operator, value} pairs that are used to match the IP protocol value byte in IP packets.

The operator byte is encoded as:
Numeric operator

e - end-of-list bit. Set in the last (op, value) pair in the list.

a - AND bit. If unset, the previous term is logically ORed with the current one. If set, the operation is a logical AND. It should be unset in the first operator byte of a sequence. The AND operator has higher priority than OR for the purposes of evaluating logical expressions.

len - length of the value field for this operand encodes 1 (00) - 4 (11) bytes. Type 3 flow component values are always encoded as single byte (len = 00).

lt - less than comparison between data and value.

gt - greater than comparison between data and value.

eq - equality between data and value.

The bits lt, gt, and eq can be combined to produce "less or equal", "greater or equal", and inequality values.

<table>
<thead>
<tr>
<th>lt</th>
<th>gt</th>
<th>eq</th>
<th>Resulting operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>true (independent of the value)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>== (equal)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>&gt; (greater than)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>&gt;= (greater than or equal)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>&lt; (less than)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>&lt;= (less than or equal)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>!= (not equal value)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>false (independent of the value)</td>
</tr>
</tbody>
</table>

Table 1: Comparison operation combinations
4.2.4. Type 4 - Port

Encoding:<type (1 octet), [op, value]+>

Defines a list of {operator, value} pairs that matches source OR destination TCP/UDP ports. This list is encoded using the numeric operator format defined in Section 4.2.3. Values are encoded as 1- or 2-byte quantities.

Port, source port, and destination port components evaluate to FALSE if the IP protocol field of the packet has a value other than TCP or UDP, if the packet is fragmented and this is not the first fragment, or if the system in unable to locate the transport header. Different implementations may or may not be able to decode the transport header in the presence of IP options or Encapsulating Security Payload (ESP) NULL [RFC4303] encryption.

4.2.5. Type 5 - Destination Port

Encoding:<type (1 octet), [op, value]+>

Defines a list of {operator, value} pairs used to match the destination port of a TCP or UDP packet. This list is encoded using the numeric operator format defined in Section 4.2.3. Values are encoded as 1- or 2-byte quantities.

4.2.6. Type 6 - Source Port

Encoding:<type (1 octet), [op, value]+>

Defines a list of {operator, value} pairs used to match the source port of a TCP or UDP packet. This list is encoded using the numeric operator format defined in Section 4.2.3. Values are encoded as 1- or 2-byte quantities.

4.2.7. Type 7 - ICMP type

Encoding:<type (1 octet), [op, value]+>

Defines a list of {operator, value} pairs used to match the type field of an ICMP packet. This list is encoded using the numeric operator format defined in Section 4.2.3. Values are encoded using a single byte.

The ICMP type specifiers evaluate to FALSE whenever the protocol value is not ICMP.
4.2.8. Type 8 - ICMP code

Encoding:<type (1 octet), [op, value]+>

Defines a list of {operator, value} pairs used to match the code field of an ICMP packet. This list is encoded using the numeric operator format defined in Section 4.2.3. Values are encoded using a single byte.

The ICMP code specifiers evaluate to FALSE whenever the protocol value is not ICMP.

4.2.9. Type 9 - TCP flags

Encoding:<type (1 octet), [op, bitmask]+>

Bitmask values can be encoded as a 1- or 2-byte bitmask. When a single byte is specified, it matches byte 13 of the TCP header [RFC0793], which contains bits 8 through 15 of the 4th 32-bit word. When a 2-byte encoding is used, it matches bytes 12 and 13 of the TCP header with the data offset field having a "don’t care" value.

This component evaluates to FALSE for packets that are not TCP packets.

This type uses the bitmask operand format, which differs from the numeric operator format in the lower nibble.

```
0 1 2 3 4 5 6 7
+---+---+---+---+---+---+---+---+
| e | a | len | 0 | 0 | not | m |
+---+---+---+---+---+---+---+---+
```

Bitmask format

e, a, len - Most significant nibble: (end-of-list bit, AND bit, and length field), as defined for in the numeric operator format in Section 4.2.3.

not - NOT bit. If set, logical negation of operation.

m - Match bit. If set, this is a bitwise match operation defined as "((data AND value) == value"; if unset, (data AND value) evaluates to TRUE if any of the bits in the value mask are set in the data
4.2.10. Type 10 - Packet length

Encoding: <type (1 octet), [op, bitmask]+>

Defines a list of {operator, value} pairs used to match on the total IP packet length (excluding Layer 2 but including IP header). This list is encoded using the numeric operator format defined in Section 4.2.3. Values are encoded using 1- or 2-byte quantities.

4.2.11. Type 11 - DSCP (Diffserv Code Point)

Encoding: <type (1 octet), [op, value]+>

Defines a list of {operator, value} pairs used to match the 6-bit DSCP field [RFC2474]. This list is encoded using the numeric operator format defined in Section 4.2.3. Values are encoded using a single byte. The two most significant bits are zero and the six least significant bits contain the DSCP value.

4.2.12. Type 12 - Fragment

Encoding: <type (1 octet), [op, bitmask]+>

Uses bitmask operand format defined in Section 4.2.9.

0 1 2 3 4 5 6 7
+------------------------+
|  Reserved  |LF |FF |IsF|DF |
+------------------------+

Bitmask values:

Bit 7 - Don’t fragment (DF)
Bit 6 - Is a fragment (IsF)
Bit 5 - First fragment (FF)
Bit 4 - Last fragment (LF)

4.3. Examples of Encodings

An example of a flow specification encoding for: "all packets to 10.0.1/24 and TCP port 25".
An example of a flow specification encoding for: "all packets to 10.1.1/24 from 192/8 and port (range [137, 139] or 8080)".

<table>
<thead>
<tr>
<th>destination</th>
<th>source</th>
<th>port</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01 18 0a 01 01</td>
<td>02 08 c0</td>
<td>04 03 89 45 8b 91 1f 90</td>
</tr>
</tbody>
</table>

Decode for port:

<table>
<thead>
<tr>
<th>Value</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0x04</td>
<td>type</td>
</tr>
<tr>
<td>0x03</td>
<td>operator</td>
</tr>
<tr>
<td>0x89</td>
<td>value</td>
</tr>
<tr>
<td>0x45</td>
<td>operator</td>
</tr>
<tr>
<td>0x8b</td>
<td>value</td>
</tr>
<tr>
<td>0x91</td>
<td>operator</td>
</tr>
<tr>
<td>0x1f90</td>
<td>value</td>
</tr>
</tbody>
</table>

This constitutes an NLRI with an NLRI length of 16 octets.

5. Traffic Filtering

Traffic filtering policies have been traditionally considered to be relatively static. Limitations of the static mechanisms caused this mechanism to be designed for the three new applications of traffic filtering (prevention of traffic-based, denial-of-service (DOS)
attacks, traffic filtering in the context of BGP/MPLS VPN service, and centralized traffic control for SDN/NFV networks) requires coordination among service providers and/or coordination among the AS within a service provider. Section 8 has details on the limitation of previous mechanisms and why BGP Flow Specification version 1 provides a solution for to prevent DOS and aid BGP/MPLS VPN filtering rules.

This flow specification NLRI defined above to convey information about traffic filtering rules for traffic that should be discarded or handled in manner specified by a set of pre-defined actions (which are defined in BGP Extended Communities). This mechanism is primarily designed to allow an upstream autonomous system to perform inbound filtering in their ingress routers of traffic that a given downstream AS wishes to drop.

In order to achieve this goal, this draft specifies two application specific NLRI identifiers that provide traffic filters, and a set of actions encoding in BGP Extended Communities. The two application specific NLRI identifiers are:

- IPv4 flow specification identifier (AFI=1, SAFI=133) along with specific semantic rules for IPv4 routes, and
- BGP NLRI type (AFI=1, SAFI=134) value, which can be used to propagate traffic filtering information in a BGP/MPLS VPN environment.

Distribution of the IPv4 Flow specification is described in section 6, and distribution of BGP/MPLS traffic flow specification is described in section 8. The traffic filtering actions are described in section 7.

5.1. Ordering of Traffic Filtering Rules

With traffic filtering rules, more than one rule may match a particular traffic flow. Thus, it is necessary to define the order at which rules get matched and applied to a particular traffic flow. This ordering function must be such that it must not depend on the arrival order of the flow specification’s rules and must be consistent in the network.

The relative order of two flow specification rules is determined by comparing their respective components. The algorithm starts by comparing the left-most components of the rules. If the types differ, the rule with lowest numeric type value has higher precedence (and thus will match before) than the rule that doesn’t contain that
component type. If the component types are the same, then a type-specific comparison is performed.

For IP prefix values (IP destination and source prefix) precedence is given to the lowest IP value of the common prefix length; if the common prefix is equal, then the most specific prefix has precedence.

For all other component types, unless otherwise specified, the comparison is performed by comparing the component data as a binary string using the memcmp() function as defined by the ISO C standard. For strings of different lengths, the common prefix is compared. If equal, the longest string is considered to have higher precedence than the shorter one.

Pseudocode:

```c
flow_rule_cmp (a, b) {
    comp1 = next_component(a);
    comp2 = next_component(b);
    while (comp1 || comp2) {
        // component_type returns infinity on end-of-list
        if (component_type(comp1) < component_type(comp2)) {
            return A_HAS_PRECEDENCE;
        }
        if (component_type(comp1) > component_type(comp2)) {
            return B_HAS_PRECEDENCE;
        }
        if (component_type(comp1) == IP_DESTINATION || IP_SOURCE) {
            common = MIN(prefix_length(comp1), prefix_length(comp2));
            cmp = prefix_compare(comp1, comp2, common);
            // not equal, lowest value has precedence
            // equal, longest match has precedence
        } else {
            common = MIN(component_length(comp1), component_length(comp2));
            cmp = memcmp(data(comp1), data(comp2), common);
            // not equal, lowest value has precedence
            // equal, longest string has precedence
        }
        return EQUAL;
    }
}
```
6. Validation Procedure

Flow specifications received from a BGP peer that are accepted in the respective Adj-RIB-In are used as input to the route selection process. Although the forwarding attributes of two routes for the same flow specification prefix may be the same, BGP is still required to perform its path selection algorithm in order to select the correct set of attributes to advertise.

The first step of the BGP Route Selection procedure (Section 9.1.2 of [RFC4271]) is to exclude from the selection procedure routes that are considered non-feasible. In the context of IP routing information, this step is used to validate that the NEXT_HOP attribute of a given route is resolvable.

The concept can be extended, in the case of flow specification NLRI, to allow other validation procedures.

A flow specification NLRI must be validated such that it is considered feasible if and only if:

a) The originator of the flow specification matches the originator of the best-match unicast route for the destination prefix embedded in the flow specification.

b) There are no more specific unicast routes, when compared with the flow destination prefix, that have been received from a different neighboring AS than the best-match unicast route, which has been determined in step a).

By originator of a BGP route, we mean either the BGP originator path attribute, as used by route reflection, or the transport address of the BGP peer, if this path attribute is not present.

BGP implementations MUST also enforce that the AS_PATH attribute of a route received via the External Border Gateway Protocol (eBGP) contains the neighboring AS in the left-most position of the AS_PATH attribute. While this rule is optional in the BGP specification, it becomes necessary to enforce it for security reasons.

The best-match unicast route may change over the time independently of the flow specification NLRI. Therefore, a revalidation of the flow specification NLRI MUST be performed whenever unicast routes change. Revalidation is defined as retesting that clause a and clause b above are true.

Explanation:
The underlying concept is that the neighboring AS that advertises the best unicast route for a destination is allowed to advertise flow-spec information that conveys a more or equally specific destination prefix. Thus, as long as there are no more specific unicast routes, received from a different neighboring AS, which would be affected by that filtering rule.

The neighboring AS is the immediate destination of the traffic described by the flow specification. If it requests these flows to be dropped, that request can be honored without concern that it represents a denial of service in itself. Supposedly, the traffic is being dropped by the downstream autonomous system, and there is no added value in carrying the traffic to it.

7. Traffic Filtering Actions

This specification defines a minimum set of filtering actions that it standardizes as BGP extended community values [RFC4360]. This is not meant to be an inclusive list of all the possible actions, but only a subset that can be interpreted consistently across the network. Additional actions can be defined as either requiring standards or as vendor specific.

Implementations SHOULD provide mechanisms that map an arbitrary BGP community value (normal or extended) to filtering actions that require different mappings in different systems in the network. For instance, providing packets with a worse-than-best-effort, per-hop behavior is a functionality that is likely to be implemented differently in different systems and for which no standard behavior is currently known. Rather than attempting to define it here, this can be accomplished by mapping a user-defined community value to platform-/network-specific behavior via user configuration.

The default action for a traffic filtering flow specification is to accept IP traffic that matches that particular rule.

This document defines the following extended communities values shown in Table 2 in the form 0x8xnn where nn indicates the sub-type. Encodings for these extended communities are described below.
Table 2: Traffic Action Extended Communities

<table>
<thead>
<tr>
<th>type</th>
<th>extended community</th>
<th>encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x8006</td>
<td>traffic-rate-bytes</td>
<td>2-byte ASN, 4-byte float</td>
</tr>
<tr>
<td>0x8007</td>
<td>traffic-action</td>
<td>bitmask</td>
</tr>
<tr>
<td>0x8008</td>
<td>redirect AS-2byte</td>
<td>2-octet AS, 4-octet value</td>
</tr>
<tr>
<td>0x8108</td>
<td>redirect IPv4</td>
<td>4-octet IPv4 address, 2-octet value</td>
</tr>
<tr>
<td>0x8208</td>
<td>redirect AS-4byte</td>
<td>4-octet AS, 2-octet value</td>
</tr>
<tr>
<td>TBD</td>
<td>traffic-rate-packets</td>
<td>2-byte ASN, 4-byte float</td>
</tr>
</tbody>
</table>

Some traffic action communities may interfere with each other. Section 7.6 of this specification provides rules for handling interference between specific types of traffic actions, and error handling based on [RFC7606]. Any additional definition of a traffic actions specified by additional standards documents or vendor documents MUST specify if the traffic action interacts with an existing traffic actions, and provide error handling per [RFC7606].

The traffic actions are processed in ascending order of the sub-type found in the BGP Extended Communities. All traffic actions are specified in transitive BGP Extended Communities.

7.1. Traffic Rate in Bytes (sub-type 0x06)

The traffic-rate-bytes extended community uses the following extended community encoding:

The first two octets carry the 2-octet id, which can be assigned from a 2-byte AS number. When a 4-byte AS number is locally present, the 2 least significant bytes of such an AS number can be used. This value is purely informational and should not be interpreted by the implementation.

The remaining 4 octets carry the maximum rate information in IEEE floating point [IEEE.754.1985] format, units being bytes per second. A traffic-rate of 0 should result on all traffic for the particular flow to be discarded.

Interferes with: Traffic Rate in packets (traffic-rate-packets). Process traffic rate in bytes (sub-type 0x06) action before traffic rate in packets action (sub-type TBD).
7.2. Traffic Rate in Packets (sub-type TBD)

The traffic-rate-packets extended community uses the same encoding as the traffic-rate-bytes extended community. The floating point value carries the maximum packet rate in packets per second. A traffic-rate-packets of 0 should result in all traffic for the particular flow to be discarded.

Interferes with: Traffic Rate in bytes (traffic-rate-bytes). Process traffic rate in bytes (sub-type 0x06) action before traffic rate in packets action (sub-type TBD).

7.3. Traffic-action (sub-type 0x07)

The traffic-action extended community consists of 6 bytes of which only the 2 least significant bits of the 6th byte (from left to right) are currently defined.

```
+---+---+---+---+---+---+---+---+
|        reserved       | S | T |
+---+---+---+---+---+---+---+---+
```

where S and T are defined as:

- T: Terminal Action (bit 47): When this bit is set, the traffic filtering engine will apply any subsequent filtering rules (as defined by the ordering procedure). If not set, the evaluation of the traffic filter stops when this rule is applied.

- S:Sample (bit 46): Enables traffic sampling and logging for this flow specification.

Interferes with: No other BGP Flow Specification traffic action in this document.

7.4. IP Redirect (sub-type 0x08)

The redirect extended community allows the traffic to be redirected to a VRF routing instance that lists the specified route-target in its import policy. If several local instances match this criteria, the choice between them is a local matter (for example, the instance with the lowest Route Distinguisher value can be elected). This extended community uses the same encoding as the Route Target extended community [RFC4360].

It should be noted that the low-order nibble of the Redirect’s Type field corresponds to the Route Target Extended Community format field.
(Type).  (See Sections 3.1, 3.2, and 4 of [RFC4360] plus Section 2 of [RFC5668].)  The low-order octet (Sub-Type) of the Redirect Extended Community remains 0x08 for all three encodings of the BGP Extended Communities (AS 2-byte, AS 4-byte, and IPv4 address).

Interferes with: All other redirect functions.  All redirect functions are mutually exclusive.  If this redirect function exists, then no other redirect functions can be processed.

7.5.  Traffic Marking (sub-type 0x09)

The traffic marking extended community instructs a system to modify the DSCP bits of a transiting IP packet to the corresponding value.  This extended community is encoded as a sequence of 5 zero bytes followed by the DSCP value encoded in the 6 least significant bits of 6th byte.

Interferes with: No other action in this document.

7.6.  Rules on Traffic Action Interference

Traffic actions may interfere with each other.  If interfering traffic actions are present for a single flow specification NLRI the entire flow specification (irrespective if there are any other non conflicting actions associated with the same flow specification) SHALL be treated as BGP WITHDRAW.

This document defines 7 traffic actions which are interfering in the following way:

1. Redirect-action-communities (0x8008, 0x8108, 0x8208):

   The three redirect-communities are mutually exclusive.  Only a single redirect community may be associated with a flow specification otherwise they are interfering.

2. All traffic-action communities (including redirect-actions):

   Multiple occurences of the same (sub-type and type) traffic-action associated with a flow specification are always interfering.

When a traffic action is defined in a standards document the handling of interaction with other/same traffic actions MUST be defined as well.  Invalid interactions between actions SHOULD NOT trigger a BGP NOTIFICATION.  All error handling for error conditions based on [RFC7606].
7.6.1. Examples

(redirect vpn-a, redirect vpn-b, traffic-rate-bytes 1Mbit/s)

Redirect vpn-a and redirect vpn-b are interfering: The BGP UPDATE is treated as WITHDRAW.

(redirect vpn-a, traffic-rate-bytes 1Mbit/s, traffic-rate-bytes 2Mbit/s)

Duplicate traffic-rate-bytes are interfering: The BGP UPDATE is treated as WITHDRAW.

(redirect vpn-a, traffic-rate-bytes 1Mbit/s, traffic-rate-packets 1000)

No interfering action communities: The BGP UPDATE is subject to further processing.

8. Dissemination of Traffic Filtering in BGP/MPLS VPN Networks

Provider-based Layer 3 VPN networks, such as the ones using a BGP/MPLS IP VPN [RFC4364] control plane, may have different traffic filtering requirements than Internet service providers. But also Internet service providers may use those VPNs for scenarios like having the Internet routing table in a VRF, resulting in the same traffic filtering requirements as defined for the global routing table environment within this document. This document proposes an additional BGP NLRI type (AFI=1, SAFI=134) value, which can be used to propagate traffic filtering information in a BGP/MPLS VPN environment.

The NLRI format for this address family consists of a fixed-length Route Distinguisher field (8 bytes) followed by a flow specification, following the encoding defined above in section x of this document. The NLRI length field shall include both the 8 bytes of the Route Distinguisher as well as the subsequent flow specification.

```
+-------------------------------+
<table>
<thead>
<tr>
<th>length (0xnn or 0xfn nn)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route Distinguisher (8 bytes)</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>NLRI value (variable)</td>
</tr>
</tbody>
</table>
+-------------------------------+
```

Flow-spec NLRI for MPLS
Propagation of this NLRI is controlled by matching Route Target extended communities associated with the BGP path advertisement with the VRF import policy, using the same mechanism as described in "BGP/MPLS IP VPNs" [RFC4364].

Flow specification rules received via this NLRI apply only to traffic that belongs to the VRF(s) in which it is imported. By default, traffic received from a remote PE is switched via an MPLS forwarding decision and is not subject to filtering.

Contrary to the behavior specified for the non-VPN NLRI, flow rules are accepted by default, when received from remote PE routers.

8.1. Validation Procedures for BGP/MPLS VPNs

The validation procedures are the same as for IPv4.

8.2. Traffic Actions Rules

The traffic action rules are the same as for IPv4.

9. Limitations of Previous Traffic Filtering Efforts

9.1. Limitations in Previous DDoS Traffic Filtering Efforts

The popularity of traffic-based, denial-of-service (DoS) attacks, which often requires the network operator to be able to use traffic filters for detection and mitigation, brings with it requirements that are not fully satisfied by existing tools.

Increasingly, DoS mitigation requires coordination among several service providers in order to be able to identify traffic source(s) and because the volumes of traffic may be such that they will otherwise significantly affect the performance of the network.

Several techniques are currently used to control traffic filtering of DoS attacks. Among those, one of the most common is to inject unicast route advertisements corresponding to a destination prefix being attacked (commonly known as remote triggered blackhole RTBH). One variant of this technique marks such route advertisements with a community that gets translated into a discard Next-Hop by the receiving router. Other variants attract traffic to a particular node that serves as a deterministic drop point.

Using unicast routing advertisements to distribute traffic filtering information has the advantage of using the existing infrastructure and inter-AS communication channels. This can allow, for instance, a
service provider to accept filtering requests from customers for address space they own.

There are several drawbacks, however. An issue that is immediately apparent is the granularity of filtering control: only destination prefixes may be specified. Another area of concern is the fact that filtering information is intermingled with routing information.

The mechanism defined in this document is designed to address these limitations. We use the flow specification NLRI defined above to convey information about traffic filtering rules for traffic that is subject to modified forwarding behavior (actions). The actions are defined as extended communities and include (but are not limited to) rate-limiting (including discard), traffic redirection, packet rewriting.

9.2. Limitations in Previous BGP/MPLS Traffic Filtering

Provider-based Layer 3 VPN networks, such as the ones using a BGP/MPLS IP VPN [RFC4364] control plane, may have different traffic filtering requirements than Internet service providers.

In these environments, the VPN customer network often has traffic filtering capabilities towards their external network connections (e.g., firewall facing public network connection). Less common is the presence of traffic filtering capabilities between different VPN attachment sites. In an any-to-any connectivity model, which is the default, this means that site-to-site traffic is unfiltered.

In circumstances where a security threat does get propagated inside the VPN customer network, there may not be readily available mechanisms to provide mitigation via traffic filter.

But also Internet service providers may use those VPNs for scenarios like having the Internet routing table in a VRF. Therefore, limitations described in Section 9.1 also apply to this section.

The BGP Flow Specification version 1 addresses these limitations.

10. Traffic Monitoring

Traffic filtering applications require monitoring and traffic statistics facilities. While this is an implementation-specific choice, implementations SHOULD provide:

- A mechanism to log the packet header of filtered traffic.
A mechanism to count the number of matches for a given flow specification rule.

11. IANA Considerations

This section complies with [RFC7153]

11.1. AFI/SAFI Definitions

For the purpose of this work, IANA has allocated values for two SAFIs: SAFI 133 for IPv4 dissemination of flow specification rules and SAFI 134 for VPNv4 dissemination of flow specification rules.

11.2. Flow Component definitions

A flow specification consists of a sequence of flow components, which are identified by a 8-bit component type. Types must be assigned and interpreted uniquely. The current specification defines types 1 through 12, with the value 0 being reserved.

IANA created and maintains a new registry entitled: "Flow Spec Component Types". The following component types have been registered:

- Type 1 - Destination Prefix
- Type 2 - Source Prefix
- Type 3 - IP Protocol
- Type 4 - Port
- Type 5 - Destination port
- Type 6 - Source port
- Type 7 - ICMP type
- Type 8 - ICMP code
- Type 9 - TCP flags
- Type 10 - Packet length
- Type 11 - DSCP
- Type 12 - Fragment
Type 13 - Bit Mask filter

In order to manage the limited number space and accommodate several usages, the following policies defined by RFC 5226 [RFC5226] are used:

<table>
<thead>
<tr>
<th>Range</th>
<th>Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Invalid value</td>
</tr>
<tr>
<td>[1 .. 12]</td>
<td>Defined by this specification</td>
</tr>
<tr>
<td>[13 .. 127]</td>
<td>Specification Required</td>
</tr>
<tr>
<td>[128 .. 255]</td>
<td>First Come First Served</td>
</tr>
</tbody>
</table>

The specification of a particular "flow component type" must clearly identify what the criteria used to match packets forwarded by the router is. This criteria should be meaningful across router hops and not depend on values that change hop-by-hop such as TTL or Layer 2 encapsulation.

The "traffic-action" extended community defined in this document has 46 unused bits, which can be used to convey additional meaning. IANA created and maintains a new registry entitled: "Traffic Action Fields". These values should be assigned via IETF Review rules only. The following traffic-action fields have been allocated:

47 Terminal Action
46 Sample
0-45 Unassigned

11.3. Extended Community Flow Specification Actions

The Extended Community Flow Specification Action types consists of two parts: BGP Transitive Extended Community types and a set of sub-types.

IANA has updated the following "BGP Transitive Extended Community Types" registries to contain the values listed below:

0x80 - Generic Transitive Experimental Use Extended Community Part 1 (Sub-Types are defined in the "Generic Transitive Experimental Extended Community Part 1 Sub-Types" Registry)
0x81 - Generic Transitive Experimental Use Extended Community Part 2 (Sub-Types are defined in the "Generic Transitive Experimental Extended Community Part 2 Sub-Types" Registry)

0x82 - Generic Transitive Experimental Use Extended Community Part 3 (Sub-Types are defined in the "Generic Transitive Experimental Use Extended Community Part 3 Sub-Types" Registry)

<table>
<thead>
<tr>
<th>RANGE</th>
<th>REGISTRATION PROCEDURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00-0xbf</td>
<td>First Come First Served</td>
</tr>
<tr>
<td>0xc0-0xff</td>
<td>IETF Review</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SUB-TYPE VALUE</th>
<th>NAME</th>
<th>REFERENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00-0x05</td>
<td>unassigned</td>
<td></td>
</tr>
<tr>
<td>0x06</td>
<td>traffic-rate</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x07</td>
<td>traffic-action</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x08</td>
<td>Flow spec redirect IPv4</td>
<td>[RFC5575] [RFC7674]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[this document]</td>
</tr>
<tr>
<td>0x09</td>
<td>traffic-marking</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x0a-0xff</td>
<td>Unassigned</td>
<td>[this document]</td>
</tr>
</tbody>
</table>

12. Security Considerations

Inter-provider routing is based on a web of trust. Neighboring autonomous systems are trusted to advertise valid reachability information. If this trust model is violated, a neighboring autonomous system may cause a denial-of-service attack by advertising reachability information for a given prefix for which it does not provide service.

As long as traffic filtering rules are restricted to match the corresponding unicast routing paths for the relevant prefixes, the security characteristics of this proposal are equivalent to the existing security properties of BGP unicast routing.

Where it is not the case, this would open the door to further denial-of-service attacks.

Enabling firewall-like capabilities in routers without centralized management could make certain failures harder to diagnose. For example, it is possible to allow TCP packets to pass between a pair of addresses but not ICMP packets. It is also possible to permit packets smaller than 900 or greater than 1000 bytes to pass between a pair of addresses, but not packets whose length is in the range 900–1000. Such behavior may be confusing and these capabilities should be used with care whether manually configured or coordinated through the protocol extensions described in this document.
13. Original authors

Barry Greene, MuPedro Marques, Jared Mauch, Danny McPherson, and Nischal Sheth were authors on [RFC5575], and therefore are contributing authors on this document.

Note: Any original author of [RFC5575] who wants to work on this draft can be added as a co-author.

14. Acknowledgements

The authors would like to thank Yakov Rekhter, Dennis Ferguson, Chris Morrow, Charlie Kaufman, and David Smith for their comments for the comments on the original [RFC5575]. Chaitanya Kodeboyina helped design the flow validation procedure; and Steven Lin and Jim Washburn ironed out all the details necessary to produce a working implementation in the original [RFC5575].

Additional acknowledgements for this document will be included here. The current authors would like to thank Alexander Mayrhofer and Nicolas Fevrier for their comments and review.

15. References

15.1. Normative References


15.2. Informative References


Authors’ Addresses

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI  48176
USA
Email: shares@ndzh.com

Robert Raszuk
Bloomberg LP
731 Lexington Ave
New York City, NY  10022
USA
Email: robert@raszuk.net

Danny McPherson
Verisign
USA
Email: dmcpherson@verisign.com
Abstract

[RFC2918] defines a route refresh capability to be exchanged between BGP speakers. BGP speakers that support this capability are advertising that they can resend the entire BGP Adj-RIB-Out on receipt of a refresh request. By supporting this capability, BGP speakers are more flexible in applying any inbound routing policy changes as they no longer have to store received routes in their unchanged form or reset the session when an inbound routing policy change occurs. The route refresh capability is advertised per AFI, SAFI combination.

There are newer AFI, SAFI types that have been introduced to BGP that support a variety of route types (e.g. IPv4/MVPN, L2VPN/EVPN). Currently, there is no way to request a subset of routes in a Route Refresh message for a given AFI, SAFI. This draft defines route refresh capability extensions that help BGP speakers to request a subset of routes for a given address family. This is expected to reduce the amount of update traffic being generated by route refresh requests as well as lessen the burden on the router servicing such requests.

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."
This Internet-Draft will expire on January 09, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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 .............................................. 3
   1.1. Use Case Examples ................................. 3
2. Requirements Language ..................................... 4
3. Route Refresh Options Capability .......................... 4
4. Route Refresh Sub-Types .................................. 4
5. Route Refresh Option format ................................ 5
6. Route Refresh Option Length ................................ 6
7. Route Refresh ID .......................................... 6
8. Route Refresh Option Flags ............................... 7
9. Route Refresh Options ..................................... 7
10. Operation ................................................. 9
11. Error Handling ........................................... 10
12. IANA Considerations ..................................... 11
13. Security Considerations ................................ 11
14. Acknowledgements ....................................... 11
15. References ............................................... 12
   15.1. Normative References ............................... 12
1. Introduction

[RFC2918] defines a route refresh capability to be exchanged between BGP speakers. BGP speakers that support this capability are advertising that they can resend the entire BGP Adj-RIB-Out on receipt of a refresh request. By supporting this capability, BGP speakers are more flexible in applying inbound routing policy changes as they no longer have to store copies of received routes in their unchanged form or reset the session when an inbound routing policy change occurs. The route refresh capability is advertised per AFI, SAFI combination.

Route refresh allows routers to dynamically request a full Adj-RIB-Out update from their peers when there’s an inbound routing policy change. This is useful because routers that mutually support this capability no longer have to flap the peering session or store an extra copy of received routes in their original form. This helps by reducing memory requirements as well as eliminating the unnecessary churn caused by session flaps. [RFC2918] does not define a way for routers to request a subset of the Adj-RIB-Out for a given AFI, SAFI.

This draft defines new extensions to route refresh that will allow requesting routers to ask for a subset of the Adj-RIB-Out for a given AFI, SAFI combination. For example, routers could ask for specific route types from those address families that support multiple route types or, they could ask for a specific prefix.

As part of the new extensions, this draft combines elements of [RFC7313] and [RFC5291] and adds a new set of options to the route refresh message that will specify filters that can be applied to limit the scope of the refresh being requested. The new option format will apply to all new option types that may be defined moving forward.

1.1. Use Case Examples

The authors acknowledge that while the extensions being proposed in this draft could potentially be addressed by Route Target Constrain described in [RFC4684] by using route targets to identify desired subset of routes, this proposal includes address families where RT Constrain extension is not supported and avoids the necessity to assign and manage the route targets per desired set of routes. The approach in this draft is intended to be a single-hop refresh only, i.e., propagation of the refreshes in a way similar to RT Constrain routes is NOT intended.
Several possible use cases are discernible today:

- The capacity to refresh routes of a certain type within an address family is needed, e.g., auto discovery routes within the EVPN AF [RFC7432].

- In VPN scenarios where RT Constrain is not supported or configured, RDs can be used.

- In BGP LS [RFC7752] cases a speaker may choose to hold only a subset of routes and depending on configuration request a subset of routes. This document could provide further filters to support those use cases.

- On changes in inbound policy, when previously configured filters have been removed, only the according subset of routes may be requested.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Route Refresh Options Capability

A BGP speaker will use the BGP Capabilities Advertisement [RFC5492] to advertise the Route Refresh Options Capability to its peers. This new capability will be advertised using the Capability code [TBD] with a capability length of 0.

By advertising the Route Refresh Options Capability to a peer, a BGP speaker indicates that it is capable of receiving and processing the route refresh options described below. This new capability can be advertised along with the Enhanced Route Refresh Capability described in [RFC7313]. However, if the Route Refresh Options Capability has been negotiated by both sides of the BGP session, then it will override the Enhanced Route Refresh Capability.

4. Route Refresh Sub-Types

[RFC7313] defines route refresh BGP message sub-types that utilize the "Reserved" field of the Route Refresh message originally defined in [RFC2918]. Currently, there are three sub-types defined and this draft proposes three additional sub-types which will be used to indicate a Route Refresh message that includes options before any ORF field of the Route Refresh message as well as BoRR and EoRR Route Refresh messages with options.
0 - Normal route refresh request [RFC2918]  
    with/without Outbound Route Filtering (ORF) [RFC5291]  
1 - Demarcation of the beginning of a route refresh  
    (BoRR) operation  
2 - Demarcation of the ending of a route refresh  
    (EoRR) operation  
+ 3 - Route Refresh request with options and optional  
    ORF [RFC5291]  
+ 4 - BoRR with options  
+ 5 - EoRR with options  
255 - Reserved

When the Route Refresh Options Capability has been negotiated by both  
sides of a BGP session, both peers MUST use message types 3, 4 and 5.  
The requesting speaker MUST use the refresh ID for all refresh  
requests including those without any options, i.e., requests for the  
full BGP Adj-RIB-Out.

The Route Refresh Request Message with options will now be formatted  
as shown below

0               1               2               3
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           A F I               |     Res.      |   S A F I     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Total Option Length        |      Refresh ID#      | Flags |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 One or more Route Refresh Options             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

5. Route Refresh Option format

[RFC2918] defines the route refresh BGP message that includes only  
the AFI, SAFI of the routes being requested. This draft proposes  
extending the basic message by including options that will indicate  
to the remote BGP speaker that a subset of the entire Adj-RIB-Out is  
being requested. The remote BGP speaker will select routes that  
match the specified options and the flag settings.

As described in the previous section, the options will be added to  
the Route Refresh message before the ORF field of the message.  
Outbound Route Filtering is described in [RFC5291]. The options will  
assume the following format
6. Route Refresh Option Length

The Option Length field will occupy the two octets immediately following the Route Refresh message containing the AFI, SAFI and subtype. The purpose of this field is to allow the BGP speaker to calculate the length of any attached ORF fields by subtracting the Option Length from the Route Refresh message length.

7. Route Refresh ID

The Refresh ID field will occupy twelve bits following the Route Refresh Options Length. It is a value assigned by the requesting BGP speaker. It MUST be a strictly monotonically increasing number per peer AFI and SAFI and will be comparable using the calculations standardized in [RFC1982]. The purpose of this field is to allow the requesting BGP speaker to correlate concurrent, overlapping refresh requests and ultimately delete correct stale routes. The Refresh ID MUST be reflected in the BoRR and EoRR messages sent by the BGP speaker servicing the refresh request.

A Refresh ID value MUST NOT be reused until an EoRR with this ID has been received by the requesting speaker or the last resort time has expired. The behavior is unspecified otherwise. More specifically, defining the interval \([ LID, HID ]\) by the values

\[
LID = \text{MAX} (\text{lowest requested Refresh ID# without BoRR}, \text{lowest received BoRR without EoRR})
\]

and

\[
HID = \text{highest requested Refresh ID#}
\]

the requesting speaker MUST only use values \(V\) where \(V > LID\) and \(V > HID\) under [RFC1982].

Value of 0 SHOULD NOT be used as Refresh ID.
The sending speaker MUST NOT reorder the BoRR messages on sending in case it received multiple requests, i.e., the BoRRs MUST follow in the same sequence as the requested Route Refresh IDs.

8. Route Refresh Option Flags

This draft defines route refresh option flags to

- specify whether the receiving BGP speaker MUST logically OR the attached options or logically AND them. When the flag is clear, the router on the receiving end SHOULD logically AND the options and only refresh routes that match all received options. If the option flag is set, the router SHOULD select routes that match using a logical OR of the options. In any case the set of routes sent between the according BoRR and EoRR MUST contain at least the logically requested set.

- indicate that the receiving BGP speaker MUST clear immediately all the received Route Refresh Requests with Options, either pending or being processed. EoRRs MUST NOT be sent. The Refresh ID# on the request is free of restrictions and MUST be set as first number in the sequence number space per [RFC1982]. The C flag MUST NOT be set on BoRR or EoRR messages and CAN be used only with refresh requests.

The precise format is indicated below

```
0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+
|  .... |C|O| R |
+-+-+-+-+-+-+-+-+-+-+-+
```

- **C**: Clear pending requests and reset Refresh ID# space.
- **O**: Use logical OR of attached options
- **R**: Reserved bits

9. Route Refresh Options

This draft introduces new options carried within the Route Refresh message as shown in the following figure

```
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7
```

The option Type is a 1 octet field that uniquely identifies individual options. The Length is a 2 octet field that contains the length of the option Value field in octets. The option Value is a variable length field that is interpreted according to the value of the option Type field.

The following types are being defined in this draft and additional types can be defined subsequently as needed:

```
+ 1 - Route Type
+ 2 - NLRI Prefix
+ 3 - Route Distinguisher Prefix
```

The Route Type option would specify a particular route type that is being requested. This option applies specifically to those AFI/SAFI combinations that support multiple route types, e.g. L2VPN/EVPN and MUST be otherwise ignored. The value field would be the route type specifying which route type was being requested. The length of the option depends on the AFI/SAFI.

The NLRI Prefix option would specify a request for all matching address prefixes with their lengths equal to or greater than the specified prefix per AFI/SAFI definitions. The value field would contain the address prefix according to the NLRI specification of the AFI/SAFI contained in the Route Refresh message. For those AFI/SAFI combinations that specify NLRI containing a type and/or RD, the value field MUST exclude the type and RD and SHOULD only include any remaining NLRI fields. If the requesting speaker expects its peer to also match the type and/or RD, the speaker CAN include the type and RD prefix options accordingly. The length field would contain the length of the value field in bits.

The Route Distinguisher prefix option would specify an RD prefix that is being requested for AFs that support it. The receiving BGP speaker would then refresh all routes in the specified AFI/SAFI that matched the requested RDs. The Value field would contain the RD, its length and the mask length of the RD prefix. This option applies specifically to those AFI/SAFI combinations that support route distinguishers and MUST be otherwise ignored.
10. Operation

A BGP speaker that understands and supports Route Refresh Options SHOULD advertise the Route Refresh Options Capability in its Open message. The following procedures for route refresh are only applicable if the BGP speaker originating the route refresh has received the route refresh options capability and supports it.

When originating a Route Refresh message, a BGP speaker SHOULD use and set these options if it wants to restrict the scope of updates being refreshed. The specific options being sent will be set according to the operator's command.

When a BGP speaker receives a route refresh message that includes any options, it MUST parse the options and strongly SHOULD use them to filter outgoing NLRIs when refreshing the Adj-RIB-Out to the requesting BGP speaker.

If a BGP speaker receives the route refresh message with the message subtype set to BoRR with options as described above, then it needs to process all the included options and MUST mark all matching routes as stale as described in [RFC7313].

If a BGP speaker receives the route refresh message with the message subtype set to EoRR with options as described above, then it needs to process all the included options and delete any remaining stale routes that match the options received with the EoRR as described in [RFC7313].

A BGP speaker responding to a route refresh request MUST set the message subtypes of the BoRR and EoRR messages so that each BoRR message has a matching EoRR message. This means a BoRR message without options SHOULD only be followed eventually by an EoRR message without options. Similarly, a BoRR message with options MUST eventually be followed by an EoRR message with the same options. If BoRR and EoRR message options do not match, the outcome is unpredictable as remaining staled routes pending a refresh may get inadvertently deleted. BGP speakers MUST NOT summarize EoRR messages by combining options in order to allow the requesting BGP speaker to uniquely identify the included sets of routes when concurrent refreshes are originated with overlapping sets of routes.

Observe that overlapping refreshes with different options are possible and in such case the according BoRR and EoRR messages are associated by using their Refresh ID#. The BGP speaker responding to the route refresh requests MAY perform the refreshes in parallel. In case of concurrent refreshes overlapping same routes, the responding speaker MUST ensure that the sent advertisements will result in
deletion of the omitted routes at the time all EoRRs have been received by the remote speaker or it MUST explicitly advertise withdrawals to correct any anomalies.

The BGP speaker requesting a refresh from its peers SHOULD maintain a locally configurable upper bound on how long it will keep matching stale routes once a BoRR has been received. Each subsequent BoRR SHOULD reset this period so that any remaining stale routes are only flushed after the last BoRR has been received in case there are multiple back-to-back refreshes being sent out and the last matching EoRR is never received or arrives too late. This is an implementation specific detail.

11. Error Handling

The handling of malformed options MUST follow the procedures mentioned in [RFC7606]. This draft obsoletes some of the error handling procedures in [RFC7313] if the Route Refresh Options Capability is sent. In addition, this draft mandates the following behavior at the receiver of the route refresh request upon detection of:

Length errors - If the message length minus the fixed-size message header is less than 4, the procedure in [RFC7313] MUST be followed. Also, if the overall length of all the options or any individual option length exceeds the total number of remaining bytes, the same procedure MUST be followed.

Option type errors - Any unknown option type CAN be ignored for AND’ed options. In case of OR’ed options the receiving speaker MUST ignore all the options and de-facto treat it as a full AFI/SAFI Adj-RIB-Out refresh. Such event SHOULD be logged in either case to notify the operator.

Option value errors - Length errors which cannot be distinguished from value field errors at the receiver are treated the same as value errors. The receiver MUST send a NOTIFICATION message with the Error Code "ROUTE-REFRESH Message Error" and the subcode of Invalid Message Length to the peer. The Data field of the NOTIFICATION message MUST contain the complete ROUTE-REFRESH message.

BoRR with unknown Refresh ID# - The receiver MUST discard all pending requests and issue a Route Refresh Request with Options. The options MUST be empty and the clear flag MUST be set to resynchronize the RIBs. "Unknown" means here a BoRR which is not in the interval

\[ \text{MAX(lowest requested Refresh ID} \text{ without BoRR, highest received BoRR}+1 \text{ respecting [RFC1982]} \],

highest requested Refresh ID# ]

EoRR with unknown Refresh ID# - Those SHOULD be ignored and a warning or error MUST be logged.

BoRR or EoRR with incorrect options - analogous to BoRR with unknown Refresh ID#.

EoRR with known Refresh ID# but without preceding BoRR - analogous to EoRR with unknown Refresh ID#. Observe that this can be caused by the peer expiring last resort timer and reusing the ID# for another request before the EoRR is received. This should be extremely unlikely given the size of the refresh ID space.

12. IANA Considerations

This draft defines a new route refresh options format for BGP Route Refresh messages.

This draft defines a new route refresh capability for BGP Route Refresh messages. We request IANA to record this capability to create a new registry under BGP Capability Codes as follows:

+74 Route Refresh Options Capability

This draft defines 3 new route refresh message subtypes for BGP Route Refresh messages. We request IANA to record these subtypes to create a new registry under BGP Route Refresh Subcodes as follows:

+ 3 - Route Refresh with options
+ 4 - BoRR with options
+ 5 - EoRR with options

13. Security Considerations

This extension to BGP does not change the underlying security issues inherent in the existing [RFC7313] and [RFC4271].

14. Acknowledgements

The authors would like to thank Anant Utgikar for initial discussions resulting in this work. John Scudder and Jeff Hass provided further comments.
15. References

15.1. Normative References


15.2. Information References


Authors' Addresses

Keyur Patel  
Cisco Systems  
821 Alder Drive  
Milpitas, CA  95035  
USA  
Email: keyupate@cisco.com

Aamod Vyavaharkar  
Cisco Systems  
821 Alder Drive  
Milpitas, CA  95035  
USA  
Email: avyavaha@cisco.com

Niloofar Fazlollahi  
Cisco Systems  
821 Alder Drive  
Milpitas, CA  95035  
USA  
Email: nifazlol@cisco.com

Tony Przygienda  
Juniper Networks  
1194 N. Mathilda Ave  
Sunnyvale, CA  94089  
USA  
Email: prz@juniper.net
Segment Routing Prefix SID extensions for BGP

draft-ietf-idr-bgp-prefix-sid-27

Abstract

Segment Routing (SR) leverages the source routing paradigm. A node steers a packet through an ordered list of instructions, called segments. A segment can represent any instruction, topological or service-based. The ingress node prepends an SR header to a packet containing a set of segment identifiers (SID). Each SID represents a topological or a service-based instruction. Per-flow state is maintained only on the ingress node of the SR domain. An SR domain is defined as a single administrative domain for global SID assignment.

This document defines an optional, transitive BGP attribute for announcing BGP Prefix Segment Identifiers (BGP Prefix-SID) information and the specification for SR-MPLS SIDs.

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.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.
Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 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 (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction .................................................. 3
2. MPLS BGP Prefix SID ........................................... 4
3. BGP Prefix-SID Attribute ...................................... 5
   3.1. Label-Index TLV .......................................... 5
   3.2. Originator SRGB TLV ..................................... 6
4. Receiving BGP Prefix-SID Attribute ............................ 8
   4.1. MPLS Dataplane: Labeled Unicast .......................... 8
5. Advertising BGP Prefix-SID Attribute ........................... 10
   5.1. MPLS Dataplane: Labeled Unicast .......................... 10
6. Error Handling of BGP Prefix-SID Attribute .................... 10
7. IANA Considerations ........................................... 11
8. Manageability Considerations ................................... 12
9. Security Considerations ....................................... 13
10. Contributors .................................................. 14
11. Acknowledgements ............................................. 14
12. References ................................................... 14
   12.1. Normative References .................................... 14
   12.2. Informative References .................................. 16
Authors’ Addresses ............................................... 17
1. Introduction

The Segment Routing (SR) architecture leverages the source routing paradigm. A segment represents either a topological instruction such as "go to prefix P following shortest path" or a service instruction. Other types of segments may be defined in the future.

A segment is identified through a Segment Identifier (SID). An SR domain is defined as a single administrative domain for global SID assignment. It may be comprised of a single Autonomous System (AS) or multiple ASes under consolidated global SID administration. Typically, the ingress node of the SR domain prepends an SR header containing segments identifiers (SIDs) to an incoming packet.

As described in [I-D.ietf-spring-segment-routing], when SR is applied to the MPLS dataplane ([I-D.ietf-spring-segment-routing-mpls]), the SID consists of a label.

[I-D.ietf-spring-segment-routing] also describes how segment routing can be applied to an IPv6 dataplane (SRv6) using an IPv6 routing header containing a stack of SR SIDs encoded as IPv6 addresses [I-D.ietf-6man-segment-routing-header]. The applicability and support for Segment Routing over IPv6 is beyond the scope of this document.

A BGP-Prefix Segment is a BGP prefix with a Prefix-SID attached. A BGP Prefix-SID is always a global SID ([I-D.ietf-spring-segment-routing]) within the SR domain and identifies an instruction to forward the packet over the Equal-Cost Multi-Path (ECMP) best-path computed by BGP to the related prefix. The BGP Prefix-SID is the identifier of the BGP prefix segment. In this document, we always refer to the BGP-Prefix segment by the BGP Prefix-SID.

This document describes the BGP extension to signal the BGP Prefix-SID. Specifically, this document defines a BGP attribute known as the BGP Prefix-SID attribute and specifies the rules to originate, receive, and handle error conditions for the attribute.

The BGP Prefix-SID attribute defined in this document can be attached to prefixes from Multiprotocol BGP IPv4/IPv6 Labeled Unicast ([RFC4760], [RFC8277]). Usage of the BGP Prefix-SID attribute for other Address Family Identifier (AFI)/ Subsequent Address Family Identifier (SAFI) combinations is not defined herein but may be specified in future specifications.

[I-D.ietf-spring-segment-routing-msdc] describes example use cases where the BGP Prefix-SID is used for the above AFI/SAFI combinations.
It should be noted that:

- A BGP Prefix-SID will be global across ASes when the interconnected ASes are part of the same SR domain. Alternatively, when interconnecting ASes, the ASBRs of each domain will have to handle the advertisement of unique SIDs. The mechanisms for such interconnection are outside the scope of the protocol extensions defined in this document.

- A BGP Prefix-SID MAY be attached to a BGP prefix. This implies that each prefix is advertised individually, reducing the ability to pack BGP advertisements (when sharing common attributes).

2. MPLS BGP Prefix SID

The BGP Prefix-SID is realized on the MPLS dataplane ([I-D.ietf-spring-segment-routing-mpls]) in the following way:

The operator assigns a globally unique label index, L_I, to a locally originated prefix of a BGP speaker N which is advertised to all other BGP speakers in the SR domain.

According to [I-D.ietf-spring-segment-routing], each BGP speaker is configured with a label block called the Segment Routing Global Block (SRGB). While [I-D.ietf-spring-segment-routing] recommends using the same SRGB across all the nodes within the SR domain, the SRGB of a node is a local property and could be different on different speakers. The drawbacks of the use case where BGP speakers have different SRGBs are documented in [I-D.ietf-spring-segment-routing] and [I-D.ietf-spring-segment-routing-msdc].

If traffic-engineering within the SR domain is required, each node may also be required to advertise topological information and Peering SIDs for each of its links and peers. This information is required to perform the explicit path computation and to express an explicit path as a list of SIDs. The advertisement of topological information and peer segments (Peer SIDs) is done through [I-D.ietf-idr-bgpls-segment-routing-epe].

If a prefix segment is to be included in an MPLS label stack, e.g., for traffic engineering purposes, the knowledge of the SRGB of the originator of the prefix is required in order to compute the local label used by the originator.

This document assumes that BGP-LS is the preferred method for collecting both peer segments (Peer SIDs) and SRGB information through [RFC7752], [I-D.ietf-idr-bgpls-segment-routing-epe], and
A BGP speaker will derive its local MPLS label L from the label index L_I and its local SRGB as described in [I-D.ietf-spring-segment-routing-mpls]. The BGP speaker then programs the MPLS label L in its MPLS dataplane as its incoming/local label for the prefix. See Section 4.1 for more details.

The outgoing label for the prefix is found in the Network Layer Reachability Information (NLRI) of the Multiprotocol BGP IPv4/IPv6 Labeled Unicast prefix advertisement as defined in [RFC8277]. The label index L_I is only used as a hint to derive the local/incoming label.

Section 3.1 of this document specifies the Label-Index TLV of the BGP Prefix-SID attribute; this TLV can be used to advertise the label index for a given prefix.

3. BGP Prefix-SID Attribute

The BGP Prefix-SID attribute is an optional, transitive BGP path attribute. The attribute type code 40 has been assigned by IANA (see Section 7).

The BGP Prefix-SID attribute is defined here to be a set of elements encoded as "Type/Length/Value" tuples (i.e., a set of TLVs). All BGP Prefix-SID attribute TLVs will start with a 1-octet type and a 2-octet length. The following TLVs are defined in this document:

- Label-Index TLV
- Originator SRGB TLV

The Label-Index and Originator SRGB TLVs are used only when SR is applied to the MPLS dataplane.

For future extensibility, unknown TLVs MUST be ignored and propagated unmodified.

3.1. Label-Index TLV

The Label-Index TLV MUST be present in the BGP Prefix-SID attribute attached to IPv4/IPv6 Labeled Unicast prefixes ([RFC8277]). It MUST
be ignored when received for other BGP AFI/SAFI combinations. The Label-Index TLV has the following format:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Type    |             Length            |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Flags              |       Label Index             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Label Index          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type is 1.
- Length: is 7, the total length in octets of the value portion of the TLV.
- RESERVED: 8-bit field. MUST be clear on transmission and MUST be ignored on reception.
- Flags: 16 bits of flags. None are defined by this document. The flag field MUST be clear on transmission and MUST be ignored on reception.
- Label Index: 32-bit value representing the index value in the SRGB space.

### 3.2. Originator SRGB TLV

The Originator SRGB TLV is an optional TLV and has the following format:
where:

- Type is 3.
- Length is the total length in octets of the value portion of the TLV: 2 + (non-zero multiple of 6).
- Flags: 16 bits of flags. None are defined in this document. Flags MUST be clear on transmission and MUST be ignored on reception.
- SRGB: 3 octets specifying the first label in the range followed by 3 octets specifying the number of labels in the range. Note that the SRGB field MAY appear multiple times. If the SRGB field appears multiple times, the SRGB consists of multiple ranges that are concatenated.

The Originator SRGB TLV contains the SRGB of the node originating the prefix to which the BGP Prefix-SID is attached. The Originator SRGB TLV MUST NOT be changed during the propagation of the BGP update. It is used to build segment routing policies when different SRGBs are used in the fabric, for example ([I-D.ietf-spring-segment-routing-msdc]).

Examples of how the receiving routers concatenate the ranges and build their neighbor’s Segment Routing Global Block (SRGB) are included in [I-D.ietf-spring-segment-routing-mpls]).
The Originator SRGB TLV may only appear in a BGP Prefix-SID attribute attached to IPv4/IPv6 Labeled Unicast prefixes ([RFC8277]). It MUST be ignored when received for other BGP AFI/SAFI combinations. Since the Label-Index TLV is required for IPv4/IPv6 prefix applicability, the Originator SRGB TLV will be ignored if it is not specified consistent with Section 6.

If a BGP speaker receives a node’s SRGB as an attribute of the BGP-LS Node NLRI and the BGP speaker also receives the same node’s SRGB in a BGP Prefix-SID attribute, then the received values should be the same. If the values are different, the values advertised in the BGP-LS NLRI SHOULD be preferred and an error should be logged.

4. Receiving BGP Prefix-SID Attribute

A BGP speaker receiving a BGP Prefix-SID attribute from an External BGP (EBGP) neighbor residing outside the boundaries of the SR domain MUST discard the attribute unless it is configured to accept the attribute from the EBGP neighbor. A BGP speaker SHOULD log an error for further analysis when discarding an attribute.

4.1. MPLS Dataplane: Labeled Unicast

A BGP session supporting the Multiprotocol BGP labeled IPv4 or IPv6 Unicast ([RFC8277]) AFI/SAFI is required.

When the BGP Prefix-SID attribute is attached to a BGP labeled IPv4 or IPv6 Unicast [RFC8277] AFI/SAFI, it MUST contain the Label-Index TLV and MAY contain the Originator SRGB TLV. A BGP Prefix-SID attribute received without a Label-Index TLV MUST be considered as "invalid" by the receiving speaker.

The label index provides guidance to the receiving BGP speaker as to the incoming label that SHOULD be allocated to the prefix.

A BGP speaker may be locally configured with an SRGB=[SRGB_Start, SRGB_End]. The preferred method for deriving the SRGB is a matter of local node configuration.

The mechanisms through which a given label index value is assigned to a given prefix are outside the scope of this document.

Given a label index \( L_I \), we refer to \( L = L_I + SRGB_{\text{Start}} \) as the derived label. A BGP Prefix-SID attribute is designated "conflicting" for a speaker \( M \) if the derived label value \( L \) lies outside the SRGB configured on \( M \). Otherwise the Label-Index TLV is designated "acceptable" to speaker \( M \).
If multiple different prefixes are received with the same label index, all of the different prefixes MUST have their BGP Prefix-SID attribute considered as "conflicting".

If multiple valid paths for the same prefix are received from multiple BGP speakers or, in the case of [RFC7911], from the same BGP speaker, and the BGP Prefix-SID attributes do not contain the same label index, then the label index from the best path BGP Prefix-SID attribute SHOULD be chosen with a notable exception being when [RFC5004] is being used to dampen route changes.

When a BGP speaker receives a path from a neighbor with an "acceptable" BGP Prefix-SID attribute and that path is selected as the best path, it SHOULD program the derived label as the label for the prefix in its local MPLS dataplane.

When a BGP speaker receives a path from a neighbor with an "invalid" or "conflicting" BGP Prefix-SID attribute or when a BGP speaker receives a path from a neighbor with a BGP Prefix-SID attribute but is unable to process it (e.g., local policy disables the functionality), it MUST ignore the BGP Prefix-SID attribute. For the purposes of label allocation, a BGP speaker MUST assign a local (also called dynamic) label (non-SRGB) for such a prefix as per classic Multiprotocol BGP IPv4/IPv6 Labeled Unicast ([RFC8277]) operation.

In the case of an "invalid" BGP Prefix-SID attribute, a BGP speaker MUST follow the error handling rules specified in Section 6. A BGP speaker SHOULD log an error for further analysis. In the case of a "conflicting" BGP Prefix-SID attribute, a BGP speaker SHOULD NOT treat it as error and SHOULD propagate the attribute unchanged. A BGP Speaker SHOULD log a warning for further analysis, i.e., in the case the conflict is not due to a label index transition.

When a BGP Prefix-SID attribute changes and transitions from "conflicting" to "acceptable", the BGP Prefix-SID attributes for other prefixes may also transition to "acceptable" as well. Implementations SHOULD assure all impacted prefixes revert to using the label indices corresponding to these newly "acceptable" BGP Prefix-SID attributes.

The outgoing label is always programmed as per classic Multiprotocol BGP IPv4/IPv6 Labeled Unicast ([RFC8277]) operation. Specifically, a BGP speaker receiving a prefix with a BGP Prefix-SID attribute and a label NLRI field of Implicit NULL [RFC3032] from a neighbor MUST adhere to standard behavior and program its MPLS dataplane to pop the top label when forwarding traffic to the prefix. The label NLRI defines the outbound label that MUST be used by the receiving node.
5. Advertising BGP Prefix-SID Attribute

The BGP Prefix-SID attribute MAY be attached to BGP IPv4/IPv6 Label Unicast prefixes [RFC8277]. In order to prevent distribution of the BGP Prefix-SID attribute beyond its intended scope of applicability, attribute filtering SHOULD be deployed to remove the BGP Prefix-SID attribute at the administrative boundary of the segment routing domain.

A BGP speaker that advertises a path received from one of its neighbors SHOULD advertise the BGP Prefix-SID received with the path without modification, as long as the BGP Prefix-SID was acceptable. If the path did not come with a BGP Prefix-SID attribute, the speaker MAY attach a BGP Prefix-SID to the path if configured to do so. The content of the TLVs present in the BGP Prefix-SID is determined by the configuration.

5.1. MPLS Dataplane: Labeled Unicast

A BGP speaker that originates a prefix attaches the BGP Prefix-SID attribute when it advertises the prefix to its neighbors via Multiprotocol BGP IPv4/IPv6 Labeled Unicast ([RFC8277]). The value of the label index in the Label-Index TLV is determined by configuration.

A BGP speaker that originates a BGP Prefix-SID attribute MAY optionally announce the Originator SRGB TLV along with the mandatory Label-Index TLV. The content of the Originator SRGB TLV is determined by configuration.

Since the label index value must be unique within an SR domain, by default an implementation SHOULD NOT advertise the BGP Prefix-SID attribute outside an Autonomous System unless it is explicitly configured to do so.

In all cases, the label field of the advertised NLRI ([RFC8277], [RFC4364]) MUST be set to the local/incoming label programmed in the MPLS dataplane for the given advertised prefix. If the prefix is associated with one of the BGP speaker’s interfaces, this is the usual MPLS label (such as the Implicit or Explicit NULL label [RFC3032]).

6. Error Handling of BGP Prefix-SID Attribute

When a BGP Speaker receives a BGP Update message containing a malformed or invalid BGP Prefix-SID attribute attached to a IPv4/IPv6 Labeled Unicast prefix [RFC8277], it MUST ignore the received BGP Prefix-SID attributes and not advertise it to other BGP peers. In
this context, a malformed BGP Prefix-SID attribute is one that cannot
be parsed due to not meeting the minimum attribute length
requirement, contains a TLV length that doesn’t conform to the length
constraints for the TLV, or a contains TLV length that would extend
beyond the end of the attribute (as defined by the attribute length).
This is equivalent to the "Attribute discard" action specified in
[RFC7606]. When discarding an attribute, a BGP speaker SHOULD log an
error for further analysis.

As per with [RFC7606], if the BGP Prefix-SID attribute appears more
than once in an UPDATE message, then all the occurrences of the
attribute other than the first one SHALL be discarded and the UPDATE
message will continue to be processed. Similarly, if a recognized
TLV appears more than once in an BGP Prefix-SID attribute while the
specification only allows for a single occurrence, then all the
occurrences of the TLV other than the first one SHALL be discarded
and the Prefix-SID attribute will continue to be processed.

For future extensibility, unknown TLVs MUST be ignored and propagated
unmodified.

7. IANA Considerations

This document defines a BGP path attribute known as the BGP Prefix-
SID attribute. This document requests IANA to assign an attribute
code type (suggested value: 40) to the BGP Prefix-SID attribute from
the BGP Path Attributes registry.

IANA temporarily assigned the following:

40 BGP Prefix-SID (TEMPORARY - registered 2015-09-30, expires
2018-09-30) [draft-ietf-idr-bgp-prefix-sid]

This document defines two TLVs for the BGP Prefix-SID attribute.
These TLVs need to be registered with IANA. We request IANA to
create a registry for BGP Prefix-SID Attribute TLVs as follows:

Under "Border Gateway Protocol (BGP) Parameters" registry, "BGP
Prefix-SID TLV Types" Reference: draft-ietf-idr-bgp-prefix-sid
Registration Procedure(s): Values 1-254 - Expert Review as defined in
[RFC8126], Value 0 and 255 reserved

Value Type            Reference
0     Reserved        this document
1     Label-Index     this document
2     Deprecated      this document
3     Originator SRGB this document
4-254 Unassigned
255   Reserved        this document

The value 2 previously corresponded to the IPv6 SID TLV which was specified in previous versions of this document. It was removed and usage of the BGP Prefix-SID for Segment Routing over the IPv6 dataplane [I-D.ietf-spring-segment-routing] has been deferred to future specifications.

This document also requests creation of the "BGP Prefix-SID Label-Index TLV Flags" registry under the "Border Gateway Protocol (BGP) Parameters" registry, Reference: draft-ietf-idr-bgp-prefix-sid. Initially, this 16-bit flags registry will be empty. The registration policy for flag bits will Expert Review [RFC8126] consistent with the BGP Prefix-SID TLV Types registry.

Finally, this document requests creation of the "BGP Prefix-SID Originator SRGB TLV Flags" registry under the "Border Gateway Protocol (BGP) Parameters" registry, Reference: draft-ietf-idr-bgp-prefix-sid. Initially, this 16-bit flags registry will be empty. The registration policy for flag bits will Expert Review [RFC8126] consistent with the BGP Prefix-SID TLV Types registry.

The designated experts must be good and faithful stewards of the above registries, assuring that each request is legitimate and corresponds to a viable use case. Given the limited number of bits in the flags registries and the applicability to a single TLV, additional scrutiny should be afforded to flag bit allocation requests. In general, no single use case should require more than one flag bit and, should the use case require more, alternate encodings using new TLVs should be considered.

8. Manageability Considerations

This document defines a BGP attribute to address use cases such as the one described in [I-D.ietf-spring-segment-routing-msdc]. It is assumed that advertisement of the BGP Prefix-SID attribute is controlled by the operator in order to:

- Prevent undesired origination/advertisement of the BGP Prefix-SID attribute. By default, a BGP Prefix-SID attribute SHOULD NOT be attached to a prefix and advertised. Hence, BGP Prefix-SID advertisement SHOULD require explicit enablement.
- Prevent any undesired propagation of the BGP Prefix-SID attribute. By default, the BGP Prefix-SID is not advertised outside the boundary of a single SR/administrative domain which may include one or more ASes. The propagation to other ASes MUST be explicitly configured.

The deployment model described in [I-D.ietf-spring-segment-routing-msdc] assumes multiple Autonomous Systems (ASes) under a common administrative domain. For this use case, the BGP Prefix-SID advertisement is applicable to the inter-AS context, i.e., EBGP, while it is confined to a single administrative domain.

9. Security Considerations

This document introduces a BGP attribute (BGP Prefix-SID) which inherits the security considerations expressed in: [RFC4271], [RFC8277], and [I-D.ietf-spring-segment-routing].

When advertised using BGPsec as described in [RFC8205], the BGP Prefix-SID attribute doesn’t impose any unique security considerations. It should be noted that the BGP Prefix-SID attribute is not protected by the BGPsec signatures.

It should be noted that, as described in Section 8, this document refers to a deployment model where all nodes are under the single administrative domain. In this context, we assume that the operator doesn’t want to leak any information related to internal prefixes and topology outside of the administrative domain. The internal information includes the BGP Prefix-SID. In order to prevent such leaking, the common BGP mechanisms (filters) are applied at the boundary of the SR/administrative domain. Local BGP attribute filtering policies and mechanisms are not standardized and, consequently, beyond the scope of this document.

To prevent a Denial-of-Service (DoS) or Distributed-Denial-of-Service (DDoS) attack due to excessive BGP updates with an invalid or conflicting BGP Prefix-SID attribute, error log message rate-limiting as well as suppression of duplicate error log messages SHOULD be deployed.

Since BGP-LS is the preferred method for advertising SRGB information, the BGP speaker SHOULD log an error if a BGP Prefix-SID attribute is received with SRGB information different from that received as an attribute of the same node’s BGP-LS Node NLRI.
10. Contributors

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

Saikat Ray  
Unaffiliated  
US  
Email: raysaikat@gmail.com

11. Acknowledgements

The authors would like to thank Satya Mohanty for his contribution to this document.

The authors would like to thank Alvaro Retana for substantive comments as part of the Routing AD review.

The authors would like to thank Bruno Decraene for substantive comments and suggested text as part of the Routing Directorate review.

The authors would like to thank Shyam Sethuram for comments and discussion of TLV processing and validation.

The authors would like to thank Robert Raszuk for comments and suggestions regarding the MPLS data plane behavior.

The authors would like to thank Krishna Deevi, Juan Alcaide, Howard Yang, and Jakob Heitz for discussions on conflicting BGP Prefix-SID label indices and BGP add paths.

The authors would like to thank Peter Yee, Tony Przygienda, Mirja Kuehlewind, Alexey Melnikov, Eric Rescorla, Suresh Krishnan, Warren Kumari, Ben Campbell, Sue Hares, and Martin Vigoureux for IDR Working Group last call, IETF Last Call, directorate, and IESG reviews.

12. References

12.1. Normative References
[I-D.ietf-spring-segment-routing]

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


12.2. Informative References

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

[I-D.ietf-idr-bgp-ls-segment-routing-ext]

[I-D.ietf-idr-bgppls-segment-routing-epe]

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

[RFC3032]

[RFC5004]

Authors' Addresses

Stefano Previdi
Cisco Systems
IT

Email: stefano@previdi.net

Clarence Filsfils
Cisco Systems
Brussels
Belgium

Email: cfilesfils@cisco.com

Acee Lindem (editor)
Cisco Systems
301 Midenhall Way
Cary, NC  27513
USA

Email: acee@cisco.com

Arjun Sreekantiah

Email: arjunhrs@gmail.com

Hannes Gredler
RtBrick Inc.

Email: hannes@rtbrick.com
BGP-LS extensions for Segment Routing BGP Egress Peer Engineering
draft-ietf-idr-bgpls-segment-routing-epe-19

Abstract

Segment Routing (SR) leverages source routing. A node steers a packet through a controlled set of instructions, called segments, by prepending the packet with an SR header. A segment can represent any instruction, topological or service-based. SR segments allow steering a flow through any topological path and service chain while maintaining per-flow state only at the ingress node of the SR domain.

This document describes an extension to BGP Link-State (BGP-LS) for advertisement of BGP Peering Segments along with their BGP peering node information so that efficient BGP Egress Peer Engineering (EPE) policies and strategies can be computed based on Segment Routing.

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.

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 November 17, 2019.

Copyright Notice

Copyright (c) 2019 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.

Table of Contents

1. Introduction .................................................. 3
2. BGP Peering Segments .......................................... 4
3. BGP-LS NLRI Advertisement for BGP Protocol .................. 5
   3.1. BGP Router-ID and Member AS Number .................... 6
   3.2. Mandatory BGP Node Descriptors ........................ 6
   3.3. Optional BGP Node Descriptors ........................... 7
4. BGP-LS Attributes for BGP Peering Segments .................. 7
   4.1. Advertisement of the PeerNode SID ....................... 10
   4.2. Advertisement of the PeerAdj SID ........................ 11
   4.3. Advertisement of the PeerSet SID ........................ 12
5. IANA Considerations ............................................ 12
   5.1. New BGP-LS Protocol-ID .................................. 13
   5.2. Node Descriptors and Link Attribute TLVs ............... 13
6. Manageability Considerations ................................... 14
7. Security Considerations ........................................ 15
8. Contributors .................................................... 15
9. Acknowledgements ............................................... 16
10. References ...................................................... 16
    10.1. Normative References ................................... 16
    10.2. Informative References ................................ 17
Authors’ Addresses ................................................ 17
1. Introduction

Segment Routing (SR) leverages source routing. A node steers a packet through a controlled set of instructions, called segments, by prepending the packet with an SR header with segment identifiers (SID). A SID can represent any instruction, topological or service-based. SR segments allows to enforce a flow through any topological path or service function while maintaining per-flow state only at the ingress node of the SR domain.

The SR architecture [RFC8402] defines three types of BGP Peering Segments that may be instantiated at a BGP node:

- Peer Node Segment (PeerNode SID) : instruction to steer to a specific peer node
- Peer Adjacency Segment (PeerAdj SID) : instruction to steer over a specific local interface towards a specific peer node
- Peer Set Segment (PeerSet SID) : instruction to load-balance to a set of specific peer nodes

SR can be directly applied to either to an MPLS dataplane (SR/MPLS) with no change on the forwarding plane or to a modified IPv6 forwarding plane (SRv6).

This document describes extensions to the BGP Link-State NLRI (BGP-LS NLRI) and the BGP-LS Attribute defined for BGP-LS [RFC7752] for advertising BGP peering segments from a BGP node along with its peering topology information (i.e., its peers, interfaces, and peering ASs) to enable computation of efficient BGP Egress Peer Engineering (BGP-EPE) policies and strategies using the SR/MPLS dataplane. The corresponding extensions for SRv6 are specified in [I-D.dawra-idr-bgpls-srv6-ext].

[I-D.ietf-spring-segment-routing-central-epe] illustrates a centralized controller-based BGP Egress Peer Engineering solution involving SR path computation using the BGP Peering Segments. This use case comprises a centralized controller that learns the BGP Peering SIDs via BGP-LS and then uses this information to program a BGP-EPE policy at any node in the domain to perform traffic steering via a specific BGP egress node to a specific EBGP peer(s) optionally also over a specific interface. The BGP-EPE policy can be realized using the SR Policy framework [I-D.ietf-spring-segment-routing-policy].

This document introduces a new BGP-LS Protocol-ID for BGP and defines new BGP-LS Node and Link Descriptor TLVs to facilitate advertising
BGP-LS Link NLRI to represent the BGP peering topology. Further, it specifies the BGP-LS Attribute TLVs for advertisement of the BGP Peering Segments (i.e., PeerNode SID, PeerAdj SID, and PeerSet SID) to be advertised in the same BGP-LS Link NLRI.

2. BGP Peering Segments

As described in [RFC8402], a BGP-EPE enabled Egress PE node instantiates SR Segments corresponding to its attached peers. These segments are called BGP Peering Segments or BGP Peering SIDs. In case of EBGP, they enable the expression of source-routed inter-domain paths.

An ingress border router of an AS may compose a list of SIDs to steer a flow along a selected path within the AS, towards a selected egress border router C of the AS, and to a specific EBGP peer. At minimum, a BGP-EPE policy applied at an ingress PE involves two SIDs: the Node SID of the chosen egress PE and then the BGP Peering SID for the chosen egress PE peer or peering interface.

Each BGP session MUST be described by a PeerNode SID. The description of the BGP session MAY be augmented by additional PeerAdj SIDs. Finally, multiple PeerNode SIDs or PeerAdj SIDs MAY be part of the same group/set in order to group EPE resources under a common PeerSet SID. These BGP Peering SIDs and their encoding are described in detail in Section 4.

The following BGP Peering SIDs need to be instantiated on a BGP router for each of its BGP peer sessions that are enabled for Egress Peer Engineering:

- One PeerNode SID MUST be instantiated to describe the BGP peer session.
- One or more PeerAdj SID MAY be instantiated corresponding to the underlying link(s) to the directly connected BGP peer session.
- A PeerSet SID MAY be instantiated and additionally associated and shared between one or more PeerNode SIDs or PeerAdj SIDs.

While an egress point in a topology usually refers to EBGP sessions between external peers, there’s nothing in the extensions defined in this document that would prevent the use of these extensions in the context of IBGP sessions. However, unlike EBGP sessions which are generally between directly connected BGP routers which are also along the traffic forwarding path, IBGP peer sessions may be setup to BGP routers which are not in the forwarding path. As such, when the IBGP design includes sessions with route-reflectors, a BGP router SHOULD
NOT instantiate a BGP Peering SID for those sessions to peer nodes which are not in the forwarding path since the purpose of BGP Peering SID is to steer traffic to that specific peers. Thus, the applicability for IBGP peering may be limited to only those deployments where the IBGP peer is also along the forwarding data path.

Any BGP Peering SIDs instantiated on the node are advertised via BGP-LS Link NLRI type as described in the sections below. An illustration of the BGP Peering SIDs’ allocations in a reference BGP peering topology along with the information carried in the BGP-LS Link NLRI and its corresponding BGP-LS Attribute are described in [I-D.ietf-spring-segment-routing-central-epe].

3. BGP-LS NLRI Advertisement for BGP Protocol

This section describes the BGP-LS NLRI encodings that describe the BGP peering and link connectivity between BGP routers.

This document specifies the advertisement of BGP peering topology information via BGP-LS Link NLRI type which requires use of a new BGP-LS Protocol-ID.

```
+-------------+----------------------------------+
| Protocol-ID | NLRI information source protocol |
+-------------+----------------------------------+
|      7      | BGP                              |
+-------------+----------------------------------+
```

Table 1: BGP-LS Protocol Identifier for BGP

The use of a new Protocol-ID allows separation and differentiation between the BGP-LS NLRIs carrying BGP information from the BGP-LS NLRIs carrying IGP link-state information defined in [RFC7752].

The BGP Peering information along with their Peering Segments are advertised using BGP-LS Link NLRI type with the Protocol-ID set to BGP. The BGP-LS Link NLRI type uses the Descriptor TLVs and BGP-LS Attribute TLVs as defined in [RFC7752]. In order to correctly describe BGP nodes, new TLVs are defined in this section.

[RFC7752] defines Link NLRI Type is as follows:
Node Descriptors and Link Descriptors are defined in [RFC7752].

3.1. BGP Router-ID and Member AS Number

Two new Node Descriptors TLVs are defined in this document:

- BGP Router Identifier (BGP Router-ID):
  - Type: 516
  - Length: 4 octets
  - Value: 4 octet unsigned non-zero integer representing the BGP Identifier as defined in [RFC6286].

- Member-AS Number (Member-ASN)
  - Type: 517
  - Length: 4 octets
  - Value: 4 octet unsigned non-zero integer representing the Member-AS Number [RFC5065].

3.2. Mandatory BGP Node Descriptors

The following Node Descriptors TLVs MUST be included in BGP-LS NLRI as Local Node Descriptors when distributing BGP information:

- BGP Router-ID (TLV 516), which contains a valid BGP Identifier of the local BGP node.
o  Autonomous System Number (TLV 512) [RFC7752], which contains the
   ASN or AS Confederation Identifier (ASN) [RFC5065], if
   confederations are used, of the local BGP node.

Note that [RFC6286] (section 2.1) requires the BGP identifier
(Router-ID) to be unique within an Autonomous System and non-zero.
Therefore, the <ASN, BGP Router-ID> tuple is globally unique. Their
use in the Node Descriptor helps map Link-State NLRIs with BGP
protocol-ID to a unique BGP router in the administrative domain where
BGP-LS is enabled.

The following Node Descriptors TLVs MUST be included in BGP-LS Link
NLRI as Remote Node Descriptors when distributing BGP information:

   o  BGP Router-ID (TLV 516), which contains the valid BGP Identifier
      of the peer BGP node.

   o  Autonomous System Number (TLV 512) [RFC7752], which contains the
      ASN or the AS Confederation Identifier (ASN) [RFC5065], if
      confederations are used, of the peer BGP node.

3.3. Optional BGP Node Descriptors

The following Node Descriptors TLVs MAY be included in BGP-LS NLRI as
Local Node Descriptors when distributing BGP information:

   o  Member-ASN (TLV 517), which contains the ASN of the confederation
      member (i.e., Member-AS Number), if BGP confederations are used,
      of the local BGP node.

   o  Node Descriptors as defined in [RFC7752].

The following Node Descriptors TLVs MAY be included in BGP-LS Link
NLRI as Remote Node Descriptors when distributing BGP information:

   o  Member-ASN (TLV 517), which contains the ASN of the confederation
      member (i.e., Member-AS Number), if BGP confederations are used,
      of the peer BGP node.

   o  Node Descriptors as defined in [RFC7752].

4. BGP-LS Attributes for BGP Peering Segments

This section defines the BGP-LS Attributes corresponding to the
following BGP Peer Segment SIDs:

   Peer Node Segment Identifier (PeerNode SID)
Peer Adjacency Segment Identifier (PeerAdj SID)

Peer Set Segment Identifier (PeerSet SID)

The following new BGP-LS Link attributes TLVs are defined for use with BGP-LS Link NLRI for advertising BGP Peering SIDs:

<table>
<thead>
<tr>
<th>TLV Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1101</td>
<td>PeerNode SID</td>
</tr>
<tr>
<td>1102</td>
<td>PeerAdj SID</td>
</tr>
<tr>
<td>1103</td>
<td>PeerSet SID</td>
</tr>
</tbody>
</table>

Figure 2: BGP-LS TLV code points for BGP-EPE

PeerNode SID, PeerAdj SID, and PeerSet SID have all the same format defined here below:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Type            |              Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Flags         |     Weight    |             Reserved          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   SID/Label/Index (variable)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: BGP Peering SIDs TLV Format

- Type: 1101, 1102 or 1103 as listed in Figure 2.
- Length: variable. Valid values are either 7 or 8 based on the whether the encoding is done as a SID Index or a label.
- Flags: one octet of flags with the following definition:

```
0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+
|V|L|B|P| Rsvd  |
+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Peering SID TLV Flags Format
* V-Flag: Value flag. If set, then the SID carries a label value. By default the flag is SET.

* L-Flag: Local Flag. If set, then the value/index carried by the SID has local significance. By default the flag is SET.

* B-Flag: Backup Flag. If set, the SID refers to a path that is eligible for protection using fast re-route (FRR). The computation of the backup forwarding path and its association with the BGP Peering SID forwarding entry is implementation specific. [I-D.ietf-spring-segment-routing-central-epe] section 3.6 discusses some of the possible ways of identifying backup paths for BGP Peering SIDs.

* P-Flag: Persistent Flag: If set, the SID is persistently allocated, i.e., the SID value remains consistent across router restart and session/interface flap.

* Rsvd bits: Reserved for future use and MUST be zero when originated and ignored when received.

  o Weight: 1 octet. The value represents the weight of the SID for the purpose of load balancing. An example use of the weight is described in [RFC8402].

  o SID/Index/Label. According to the TLV length and to the V and L flags settings, it contains either:

    * A 3 octet local label where the 20 rightmost bits are used for encoding the label value. In this case, the V and L flags MUST be SET.

    * A 4 octet index defining the offset in the Segment Routing Global Block (SRGB) [RFC8402] advertised by this router. In this case, the SRGB MUST be advertised using the extensions defined in [I-D.ietf-idr-bgp-ls-segment-routing-ext].

The values of the PeerNode SID, PeerAdj SID, and PeerSet SID Sub-TLVs SHOULD be persistent across router restart.

When enabled for Egress Peer Engineering, the BGP router MUST include the PeerNode SID TLV in the BGP-LS Attribute for the BGP-LS Link NLRI corresponding to its BGP peering sessions. The PeerAdj SID and PeerSet SID TLVs MAY be included in the BGP-LS Attribute for the BGP-LS Link NLRI.

Additional BGP-LS Link Attribute TLVs, as defined in [RFC7752] MAY be included with the BGP-LS Link NLRI in order to advertise the
characteristics of the peering link. E.g., one or more interface addresses (TLV 259 or TLV 261) of the underlying link(s) over which a multi-hop BGP peering session is setup may be included in the BGP-LS Attribute along with the PeerNode SID TLV.

4.1. Advertisement of the PeerNode SID

The PeerNode SID TLV includes a SID associated with the BGP peer node that is described by a BGP-LS Link NLRI as specified in Section 3.

The PeerNode SID, at the BGP node advertising it, has the following semantics (as defined in [RFC8402]):

- SR operation: NEXT.
- Next-Hop: the connected peering node to which the segment is associated.

The PeerNode SID is advertised with a BGP-LS Link NLRI, where:

- Local Node Descriptors include:
  - Local BGP Router-ID (TLV 516) of the BGP-EPE enabled egress PE.
  - Local ASN (TLV 512).

- Remote Node Descriptors include:
  - Peer BGP Router-ID (TLV 516) (i.e., the peer BGP ID used in the BGP session)
  - Peer ASN (TLV 512).

- Link Descriptors include the addresses used by the BGP session encoded using TLVs as defined in [RFC7752]:
  - IPv4 Interface Address (TLV 259) contains the BGP session IPv4 local address.
  - IPv4 Neighbor Address (TLV 260) contains the BGP session IPv4 peer address.
  - IPv6 Interface Address (TLV 261) contains the BGP session IPv6 local address.
  - IPv6 Neighbor Address (TLV 262) contains the BGP session IPv6 peer address.
4.2. Advertisement of the PeerAdj SID

The PeerAdj SID TLV includes a SID associated with the underlying link to the BGP peer node that is described by a BGP-LS Link NLRI as specified in Section 3.

The PeerAdj SID, at the BGP node advertising it, has the following semantics (as defined in [RFC8402]):

- SR operation: NEXT.
- Next-Hop: the interface peer address.

The PeerAdj SID is advertised with a BGP-LS Link NLRI, where:

- Local Node Descriptors include:
  - Local BGP Router-ID (TLV 516) of the BGP-EPE enabled egress PE.
  - Local ASN (TLV 512).

- Remote Node Descriptors include:
  - Peer BGP Router-ID (TLV 516) (i.e., the peer BGP ID used in the BGP session).
  - Peer ASN (TLV 512).

- Link Descriptors MUST include the following TLV, as defined in [RFC7752]:
  - Link Local/Remote Identifiers (TLV 258) contains the 4-octet Link Local Identifier followed by the 4-octet Link Remote Identifier. The value 0 is used by default when the link remote identifier is unknown.

- Additional Link Descriptors TLVs, as defined in [RFC7752], MAY also be included to describe the addresses corresponding to the link between the BGP routers:
  - IPv4 Interface Address (Sub-TLV 259) contains the address of the local interface through which the BGP session is established.
IPv6 Interface Address (Sub-TLV 261) contains the address of the local interface through which the BGP session is established.

IPv4 Neighbor Address (Sub-TLV 260) contains the IPv4 address of the peer interface used by the BGP session.

IPv6 Neighbor Address (Sub-TLV 262) contains the IPv6 address of the peer interface used by the BGP session.

- Link Attribute TLVs include the PeerAdj SID TLV as defined in Figure 3.

4.3. Advertisement of the PeerSet SID

The PeerSet SID TLV includes a SID that is shared amongst BGP peer nodes or the underlying links that are described by BGP-LS Link NLRI as specified in Section 3.

The PeerSet SID, at the BGP node advertising it, has the following semantics (as defined in [RFC8402]):

- SR operation: NEXT.
- Next-Hop: load balance across any connected interface to any peer in the associated peer set.

The PeerSet SID TLV containing the same SID value (encoded as defined in Figure 3) is included in the BGP-LS Attribute for all of the BGP-LS Link NLRI corresponding to the PeerNode or PeerAdj segments associated with the peer set.

5. IANA Considerations

This document defines:

- A new Protocol-ID: BGP. The codepoint is from the "BGP-LS Protocol-IDs" registry.

- Two new TLVs: BGP-Router-ID and BGP Confederation Member. The codepoints are in the "BGP-LS Node Descriptor, Link Descriptor, Prefix Descriptor, and Attribute TLVs" registry.

- Three new BGP-LS Attribute TLVs: PeerNode SID, PeerAdj SID and PeerSet SID. The codepoints are in the "BGP-LS Node Descriptor, Link Descriptor, Prefix Descriptor, and Attribute TLVs" registry.
5.1. New BGP-LS Protocol-ID

This document defines a new value in the registry "BGP-LS Protocol-IDs":

+------------------------------------------------------+
<table>
<thead>
<tr>
<th>Codepoint</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>BGP</td>
<td>Early Allocation by IANA</td>
</tr>
</tbody>
</table>
+------------------------------------------------------+

Figure 5: BGP Protocol Codepoint

5.2. Node Descriptors and Link Attribute TLVs

This document defines 5 new TLVs in the registry "BGP-LS Node Descriptor, Link Descriptor, Prefix Descriptor, and Attribute TLVs":

- Two new node descriptor TLVs
- Three new link attribute TLVs

All the new 5 codepoints are in the same registry: "BGP-LS Node Descriptor, Link Descriptor, Prefix Descriptor, and Attribute TLVs".

The following new Node Descriptors TLVs are defined:

+-----------------------------------------------+
<table>
<thead>
<tr>
<th>Codepoint</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>516</td>
<td>BGP Router-ID</td>
<td>Early Allocation by IANA</td>
</tr>
<tr>
<td>517</td>
<td>BGP Confederation Member</td>
<td>Early Allocation by IANA</td>
</tr>
</tbody>
</table>
+-----------------------------------------------+

Figure 6: BGP-LS Descriptor TLVs Codepoints

The following new Link Attribute TLVs are defined:

+-----------------------------------------------+
<table>
<thead>
<tr>
<th>Codepoint</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1101</td>
<td>PeerNode SID</td>
<td>Early Allocation by IANA</td>
</tr>
<tr>
<td>1102</td>
<td>PeerAdj SID</td>
<td>Early Allocation by IANA</td>
</tr>
<tr>
<td>1103</td>
<td>PeerSet SID</td>
<td>Early Allocation by IANA</td>
</tr>
</tbody>
</table>
+-----------------------------------------------+

Figure 7: BGP-LS Attribute TLVs Codepoints
6. Manageability Considerations

The new protocol extensions introduced in this document augment the existing IGP topology information BGP-LS distribution [RFC7752] by adding support for distribution of BGP peering topology information. As such, the Manageability Considerations section of [RFC7752] applies to these new extensions as well.

Specifically, the malformed Link-State NLRI and BGP-LS Attribute tests for syntactic checks in the Fault Management section of [RFC7752] now apply to the TLVs defined in this document. The semantic or content checking for the TLVs specified in this document and their association with the BGP-LS NLRI types or their associated BGP-LS Attributes is left to the consumer of the BGP-LS information (e.g., an application or a controller) and not the BGP protocol.

A consumer of the BGP-LS information retrieves this information from a BGP Speaker, over a BGP-LS session (refer Section 1 and 2 of [RFC7752]). The handling of semantic or content errors by the consumer would be dictated by the nature of its application usage and hence is beyond the scope of this document. It may be expected that an error detected in the NLRI descriptor TLVs would result in that specific NLRI update being unusable and hence its update to be discarded along with an error log. While an error in Attribute TLVs would result in only that specific attribute being discarded with an error log.

The operator MUST be provided with the options of configuring, enabling, and disabling the advertisement of each of the PeerNode SID, PeerAdj SID, and PeerSet SID as well as control of which information is advertised to which internal or external peer. This is not different from what is required by a BGP speaker in terms of information origination and advertisement.

BGP Peering Segments are associated with the normal BGP routing peering sessions. However, the BGP peering information along with these Peering Segments themselves are advertised via a distinct BGP-LS peering session. It is expected that this isolation as described in [RFC7752] is followed when advertising BGP peering topology information via BGP-LS.

BGP-EPE functionality enables the capability for instantiation of an SR path for traffic engineering a flow via an egress BGP router to a specific peer, bypassing the normal BGP best path routing for that flow and any routing policies implemented in BGP on that egress BGP router. As with any traffic engineering solution, the controller or application implementing the policy needs to ensure that there is no looping or mis-routing of traffic. Traffic counters corresponding to
the MPLS label of the BGP Peering SID on the router would indicate the traffic being forwarded based on the specific EPE path. Monitoring these counters and the flows hitting the corresponding MPLS forwarding entry would help identify issues, if any, with traffic engineering over the EPE paths. Errors in the encoding or decoding of the SR information in the TLVs defined in this document may result in the unavailability of such information to a Centralized EPE Controller or incorrect information being made available to it. This may result in the controller not being able to perform the desired SR based optimization functionality or to perform it in an unexpected or inconsistent manner. The handling of such errors by applications like such a controller may be implementation specific and out of scope of this document.

7. Security Considerations

[RFC7752] defines BGP-LS NLRI to which the extensions defined in this document apply. The Security Considerations section of [RFC7752] also applies to these extensions. The procedures and new TLVs defined in this document, by themselves, do not affect the BGP-LS security model discussed in [RFC7752].

BGP-EPE enables engineering of traffic when leaving the administrative domain via an egress BGP router. Therefore precaution is necessary to ensure that the BGP peering information collected via BGP-LS is limited to specific consumers in a secure manner. Segment Routing operates within a trusted domain [RFC8402] and its security considerations also apply to BGP Peering Segments. The BGP-EPE policies are expected to be used entirely within this trusted SR domain (e.g., between multiple AS/domains within a single provider network).

The isolation of BGP-LS peering sessions is also required to ensure that BGP-LS topology information (including the newly added BGP peering topology) is not advertised to an external BGP peering session outside an administrative domain.

8. Contributors

Mach (Guoyi) Chen
Huawei Technologies
China

Email: mach.chen@huawei.com
9. Acknowledgements

The authors would like to thank Jakob Heitz, Howard Yang, Hannes Gredler, Peter Psenak, Arjun Sreekantiah and Bruno Decraene for their feedback and comments. Susan Hares helped in improving the clarity of the document with her substantial contributions during her shepherd’s review. The authors would also like to thank Alvaro Retana for his extensive review and comments which helped correct issues and improve the document.

10. References
10.1. Normative References

[I-D.ietf-idr-bgp-ls-segment-routing-ext]


10.2. Informative References

[I-D.dawra-idr-bgpls-srv6-ext]
Dawra, G., Filsfils, C., Talaulikar, K., Chen, M.,
daniel.bernier@bell.ca, d., Uttaro, J., Decraene, B., and
H. Elmalky, "BGP Link State Extensions for SRv6", draft-
dawra-idr-bgpls-srv6-ext-06 (work in progress), March
2019.

[I-D.ietf-spring-segment-routing-central-epe]
Filsfils, C., Previdi, S., Dawra, G., Aries, E., and D.
Afanasiev, "Segment Routing Centralized BGP Egress Peer
Engineering", draft-ietf-spring-segment-routing-central-
epe-10 (work in progress), December 2017.

[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-03 (work in progress), May 2019.
Clarence Filsfils
Cisco Systems, Inc.
Brussels
Belgium

Email: cfil@ci.com

Keyur Patel
Arrcus, Inc.

Email: Keyur@arrcus.com

Saikat Ray
Individual Contributor

Email: raysaikat@gmail.com

Jie Dong
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: jie.dong@huawei.com
Flowspec Indirection-id Redirect
draft-ietf-idr-flowspec-path-redirect-11

Abstract

This document defines a new extended community known as "FlowSpec Redirect to indirection-id Extended Community". This extended community triggers advanced redirection capabilities to flowspec clients. When activated, this flowspec extended community is used by a flowspec client to retrieve the corresponding next-hop and encoding information within a localised indirection-id mapping table.

The functionality detailed in this document allows a network controller to decouple the BGP flowspec redirection instruction from the operation of the available paths.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

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 November 27, 2020.
1. Introduction

Flowspec is an extension to BGP that allows for the dissemination of traffic flow specification rules. This has many possible applications but the primary one for many network operators is the distribution of traffic filtering actions for DDoS mitigation. The flowspec standard rfc5575bis [3] defines a redirect-to-VRF action for policy-based forwarding, but this mechanism is not always sufficient, particularly if the redirected traffic needs to be steered onto an explicit path.

Every flowspec policy route is effectively a rule, consisting of two parts. The first part, encoded in the NLRI field, provides
information about the traffic matching the policy rule. The second part, encoded in one or more BGP extended communities, provides policy instructions for traffic handling on the flowspec client. The flowspec standard rfc5575bis [3] defines widely-used filter actions such as discard and rate limit; it also defines a redirect-to-VRF action for policy-based forwarding. Using the redirect-to-VRF action to steer traffic towards an alternate destination is useful for DDoS mitigation, however using this methodology can be cumbersome when there is need to steer the traffic onto an explicitly defined traffic path.

This draft specifies a "Redirect to indirection-id" flowspec action making use of a 32-bit indirection-id using a new extended community. Each indirection-id serves as anchor point, for policy-based forwarding onto an explicit path by a flowspec client.

2. indirection-id and indirection-id table

The indirection-id is a 32-bit unsigned number, used as anchor point on a flowspec client for policy-based forwarding onto an explicit path by a flowspec client.

The indirection-id table is the table construct of indirection-id values, grouped by indirection-id "ID-Type". Each entry in this table contains policy-based forwarding and encoding instructions.

The configuration of the indirection-id table on a flowspec client is a localised operation on each router, and MAY happen out-of-band from BGP flowspec. For some use-case scenarios the indirection-id "ID-Type" provides additional (maybe even fully sufficient) context for a flowspec client for policy based forwarding, making a localised indirection-id table obsolete. For example, when the indirection-id refers to a MPLS segment routing node-id [6], then the indirection-id provides sufficient information for a segment routing lookup on the flowspec client.

3. Use Case Scenarios

This section describes a few use-case scenarios when deploying "Redirect to indirection-id".

3.1. Redirection shortest Path tunnel

Description:

The first use-case describes an example where a single flowspec route is sent from a BGP flowspec controller to many BGP flowspec clients. This BGP flowspec route carries the "Redirect to indirection-id" to
all flowspec clients with intent to redirect matching dataflows onto a shortest-path tunnel pointing towards a single remote destination.

In this first use-case scenario, each flowspec client receives flowspec routes. The received flowspec routes have the extended "Redirect to indirection-id" community attached. Each "Redirect to indirection-id" community embeds two relevant components: (1) 32-bit indirection-id and (2) ID-type. These two components provide the flowspec client with sufficient information for policy based forwarding, with intent to steer and encapsulate the data-packet accordingly upon a shortest path tunnel to a single remote end-point.

Requirements:

For redirect to shortest path tunnel it is required that the tunnel MUST be operational and allow packets to flow between tunnel head- and tail-end.

Example: Indirection-ID community "ID-Type" which can be used:

- 0 (localised ID): When the intent is to use a localised Indirection-id table, configured through out-of-band procedures.

- 1 or 2 (Node ID’s): This type can be used when the goal is to use MPLS based Segment Routing towards a remote destination. In this use-case scenario the flowspec rule contains a SR (Segment Routing) node SID to steer traffic towards.

3.2. Redirection to path-engineered tunnels

Description:

The second use-case describes an example where a single flowspec route is sent from a BGP flowspec controller to many BGP flowspec clients. This BGP flowspec route carries policy information to steer traffic upon a path-engineered tunnel. It is assumed that the path engineered tunnels are configured using out-of-band from BGP flowspec.

Segment Routing Example:

For this example the indirection-id "ID-Type" points towards a Segment Routing Binding SID. The Binding SID is a segment identifier value (as per segment routing definitions in [I-D.draft-ietf-spring-segment-routing] [6]) used to associate an explicit path. The Binding SID and the associated path engineered tunnel may for example be setup by a controller using BGP as specified in [I-D.sreekantiah-idr-segment-routing-te] [5] or alternately by using PCEP as detailed
in draft-ietf-pce-segment-routing [7]. To conclude, when a BGP
speaker at some point in time receives a flowspec route with an
extended "Redirect to indirection-id’ community, it installs a
policy-based forwarding rule to redirect packets onto an explicit
path, associated with the corresponding Binding SID. The encoding of
the Binding SID within the "Redirect to indirection-id" extended
community is specified in section 4.

Requirements:

For redirect to path engineered tunnels it is required that the
tunnel MUST be operational and allow packets to flow over the
engineered path between tunnel head- and tail-end.

Example: Indirection-ID community "ID-Type" to be used:

- 0 (localised ID): When the intent is to policy-based steer traffic
using Indirection. The engineered path is configured through out-
of-band procedures and uses the 32-bit Indirection-id as local
anchor point on the local flowspec client.

- 3 or 4 (Binding Segment ID's): This type can be used when the goal
is to use MPLS based Segment Routing towards an out-of-band
configured explicit path.

- 5 (Tunnel ID): When the intent is to policy-based steer traffic
using a global tunnel-id. The engineered path is configured
through out-of-band procedures and uses the 32-bit Indirection-id
as global anchor point on the local flowspec client.

3.3. Redirection to complex dynamically constructed tunnels

Description:

A third use-case describes the application and redirection towards
complex dynamically constructed tunnels. For this use-case a BGP
flowspec controller injects a single flowspec route with two unique
"Redirect to indirection-id" communities attached, each community
tagged with a different Sequence-ID (S-ID). A flowspec client should
use the Sequence-ID (S-ID) to sequence the received flowspec redirect
information. A potential use-case scenario would for example be the
dynamic construction of Segment Routing Central Egress Path
Engineered tunnel [4] or next-next-hop tunnels.

Segment Routing Example:

i.e. a classic Segment Routing example using complex tunnels is found
in DDoS mitigation and traffic offload. Suspicious traffic (e.g.
dirty traffic flows) may be policy-based routed into a purpose built Segment Routing Central Egress Path Engineered tunnel [4]. For this complex dynamic redirect tunnel construct, a first "Redirect to indirection-id" (i.e. S-ID=0) may be used to redirect traffic into a tunnel towards a particular egress router, while a second "Redirect to indirection-id" (i.e. S-ID=1) is used to steer traffic beyond the particular egress router towards a pre-identified interface/peer. From data-plane perspective, the principles documented by [4] are valid for this use case scenario.

Requirements:

To achieve redirection towards complex dynamically constructed tunnels, multiple "Redirect to indirection-id" communities are imposed upon the flowspec route. The "Redirect to indirection-id" communities should be sequenced using the Sequence ID (S-ID). For redirect to complex dynamic engineered tunnels the tunnel MUST be operational and allow packets to flow over the engineered path between tunnel head- and tail-end.

Example: Indirection-ID community "ID-Type" to be used:

o 0 (localised ID) with S-ID: When the intent is to construct a dynamic engineered tunnel, then a sequence of localised indirection-ids may be used. The Sequence ID (S-ID) MUST be used to sequence multiple "Redirect to indirection-id" actions to construct a more complex engineered tunnel. The creation of the localised indirection-id table is operationalised out-of-band and is outside scope of this document.

4. Redirect to indirection-id Community

This document defines a new transitive BGP extended community known as "FlowSpec Redirect to indirection-id Extended Community" with the Type and the Sub-Type field to be assigned by IANA. The format of this extended community is show in Figure 1.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type          | Sub-Type      | Flags(1 octet)|    ID-Type    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Generalized indirection_id                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1

Van de Velde, et al. Expires November 27, 2020
The meaning of the extended community fields are as follows:

Type: 1 octet to be assigned by IANA.
Sub-Type: 1 octet to be assigned by IANA.
Flags: 1 octet field. Following Flags are defined.

```
+---+-+-+-+
| RES | S-ID | C |
+---+-+-+-+
```

Figure 2

The least-significant Flag bit is defined as the 'C' (or copy) bit. When the 'C' bit is set the redirection applies to copies of the matching packets and not to the original traffic stream.

The 'S-ID' field identifies a 4 bit Sequence ID field. This field is used to provide a flowspec client an indication how and where to sequence the received indirection-ids. The Sequence ID value 0 indicates that Sequence ID field is NOT set and all other sequence ID’s SHOULD be ignored. A single flowspec rule MUST NOT have more as one indirection-id per S-ID. On a flowspec client the indirection-id with lowest S-ID MUST be imposed first for any given flowspec entry.

All bits other than the 'C' and 'S-ID' bits MUST be set to 0 by the originating BGP speaker and ignored by receiving BGP speakers.

ID-Type: 1 octet value. This draft defines following Context Types:

0 - Localised ID (The flowspec client uses the received 32-bit indirection-id to lookup forwarding information within the localised indirection-id table. The allocation and programming of the localised indirection-id table is outside scope of the document)

1 - Node ID with SID/index in MPLS-based Segment Routing (This means the 32-bit indirection-id is mapped to an MPLS label using the index as a global offset in the SID/label space)
2 - Node ID with SID/label in MPLS-based Segment Routing (This means the 32-bit indirection-id is mapped to an MPLS label using the 32-bit indirection-id as global label)

3 - Binding Segment ID with SID/index in MPLS-based Segment Routing (This means the 32-bit indirection-id is mapped to an MPLS binding label using the indirection-id as index for global offset in the SID/label space) [I-D.draft-ietf-spring-segment-routing] [6]

4 - Binding Segment ID with SID/label in MPLS-based Segment Routing (This means 32-bit indirection-id is mapped to an MPLS binding label using the 32-bit indirection-id as global label) [I-D.draft-ietf-spring-segment-routing] [6]

5 - Tunnel ID (Tunnel ID is within a single administrative domain a 32-bit globally unique tunnel identifier. The allocation and programming of the Tunnel ID within the localised indirection-id table is outside scope of the document)

Generalized indirection_id: 32-bit identifier used as indirection_id

5. Redirect using localised indirection-id mapping table

When a BGP flowspec client receives a flowspec policy route with a "Redirect to indirection-id" extended community attached, and the route represents the best BGP path, it will install a flowspec policy-based forwarding rule matching the tuples described by the flowspec NLRI field and consequently redirects the flow (C=0) or copies the flow (C=1) using the information identified by the "Redirect to indirection-id" community.

6. Validation Procedures

The validation check described in rfc5575bis [3] SHOULD be applied by default by a flowspec client, for received flowspec policy routes containing a "Redirect to indirection-id" extended community. This results that a flowspec route with a destination prefix subcomponent SHOULD NOT be accepted from an EBGP peer unless that peer also advertised the best path for the matching unicast route.

While it MUST NOT happen, and is seen as invalid combination, it is possible from a semantics perspective to have multiple clashing redirect actions defined within a single flowspec rule. For best and consistent compatibility with legacy implementations, the redirect functionality as documented by rfc5575bis MUST NOT be broken, and hence when a clash occurs, then rfc5575bis based redirect MUST take priority. Additionally, if the "Redirect to indirection-id" does not
result in a valid redirection, then the flowspec rule MUST be processed as if the "Redirect to indirection-id" community was not attached to the flowspec route. In addition the flowspec client MUST provide an indication that the respective "Redirect to indirection-id" resulted in an invalid redirection action.

7. Security Considerations

A system using "Redirect to indirection-id" extended community can cause during the redirect mitigation of a DDoS attack overflow of traffic received by the mitigation infrastructure.

8. Acknowledgements

This document received valuable comments and input from IDR working group including Adam Simpson, Mustapha Aissaoui, Jan Mertens, Robert Raszuk, Jeff Haas, Susan Hares and Lucy Yong.

9. Contributors

The following people contributed to the content of this document and should be considered as co-authors:
10. IANA Considerations

This document requests a new Transitive Extended Community Type and a new registry sub-type. The new Transitive Extended Community Type name shall be "FlowSpec Redirect to indirection-id Extended Community (Sub-Types are defined in the "FlowSpec Redirect to indirection-id Extended Community Sub-Type" registry)". The name of the new Sub-type registry shall be "FlowSpec Redirect to indirection-id Extended Community Sub-Type"

Under "Transitive Extended Community:"

Registry: "FlowSpec Redirect to indirection-id Extended Community (Sub-Types are defined in the "FlowSpec Redirect to indirection-id Extended Community Sub-Type" registry)"
Registration Procedure(s): First Come, First Served

0x09 FlowSpec Redirect to indirection-id Extended Community

New Sub-Type Registry: "FlowSpec Redirect to indirection-id Extended Community Sub-Type"

<table>
<thead>
<tr>
<th>Value</th>
<th>Code</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Flowspec Redirect to 32-bit Path-id</td>
<td>[RFC-To-Be]</td>
</tr>
</tbody>
</table>

Figure 4

11. References

11.1. Normative References


11.2. Informative References


Authors' Addresses

Gunter Van de Velde (editor)
Nokia
Antwerp
BE

Email: gunter.van_de_velde@nokia.com

Keyur Patel
Arrcus
USA

Email: keyur@arrcus.com

Zhenbin Li
Huawei Technologies
Huawei Bld., No. 156 Beiquing Rd
Beijing  100095
China

Email: lizhenbin@huawei.com
Abstract

This document describes a mechanism to collect the Traffic Engineering and Policy information that is locally available in a node and advertise it into BGP Link State (BGP-LS) updates. Such information can be used by external components for path computation, re-optimization, service placement, network visualization, etc.

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 October 26, 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction ................................................................. 3
1.1. Requirements Language ............................................... 5
2. Carrying TE Policy Information in BGP ............................... 5
3. TE Policy NLRI ................................................................. 6
4. TE Policy Descriptors .................................................... 7
  4.1. Tunnel Identifier (Tunnel ID) ....................................... 8
  4.2. LSP Identifier (LSP ID) ............................................... 8
  4.3. IPv4/IPv6 Tunnel Head-End Address ............................... 9
  4.4. IPv4/IPv6 Tunnel Tail-End Address ............................... 9
  4.5. SR Policy Candidate Path Descriptor ............................. 10
  4.6. Local MPLS Cross Connect ......................................... 11
    4.6.1. MPLS Cross Connect Interface ............................... 13
    4.6.2. MPLS Cross Connect FEC .................................... 14
5. MPLS-TE Policy State TLV .............................................. 15
  5.1. RSVP Objects .......................................................... 16
  5.2. PCEP Objects .......................................................... 17
6. SR Policy State TLVs .................................................... 18
  6.1. SR Binding SID ....................................................... 18
  6.2. SRv6 Binding SID ..................................................... 20
  6.3. SR Candidate Path State .......................................... 22
  6.4. SR Policy Name ........................................................ 24
  6.5. SR Candidate Path Name ........................................... 24
  6.6. SR Candidate Path Constraints ................................... 25
    6.6.1. SR Affinity Constraint ..................................... 27
    6.6.2. SR SRLG Constraint .......................................... 28
    6.6.3. SR Bandwidth Constraint .................................... 28
    6.6.4. SR Disjoint Group Constraint ................................ 29
  6.7. SR Segment List ....................................................... 31
  6.8. SR Segment ............................................................. 33
    6.8.1. Segment Descriptors ........................................ 35
  6.9. SR Segment List Metric ............................................ 42
7. Procedures ................................................................. 44
8. Manageability Considerations ......................................... 44
9. IANA Considerations .................................................... 45
  9.1. BGP-LS NLRI-Types .................................................. 45
  9.2. BGP-LS Protocol-IDs ................................................ 45
1. Introduction

In many network environments, traffic engineering (TE) policies are instantiated into various forms:

- MPLS Traffic Engineering Label Switched Paths (TE-LSPs).
- IP based tunnels (IP in IP, GRE, etc).
- Segment Routing (SR) Policies as defined in [I-D.ietf-spring-segment-routing-policy]
- Local MPLS cross-connect configuration

All this information can be grouped into the same term: TE Policies. In the rest of this document we refer to TE Policies as the set of information related to the various instantiation of polices: MPLS TE LSPs, IP tunnels (IPv4 or IPv6), SR Policies, etc.

TE Policies are generally instantiated at the head-end and are based on either local configuration or controller based programming of the node using various APIs and protocols, e.g., PCEP or BGP.

In many network environments, the configuration and state of each TE Policy that is available in the network is required by a controller which allows the network operator to optimize several functions and operations through the use of a controller aware of both topology and state information.

One example of a controller is the stateful Path Computation Element (PCE) [RFC8231], which could provide benefits in path reoptimization. While some extensions are proposed in Path Computation Element Communication Protocol (PCEP) for the Path Computation Clients (PCCs) to report the LSP states to the PCE, this mechanism may not be...
applicable in a management-based PCE architecture as specified in section 5.5 of [RFC4655]. As illustrated in the figure below, the PCC is not an LSR in the routing domain, thus the head-end nodes of the TE-LSPs may not implement the PCEP protocol. In this case a general mechanism to collect the TE-LSP states from the ingress LERs is needed. This document proposes an TE Policy state collection mechanism complementary to the mechanism defined in [RFC8231].

![Figure 1. Management-Based PCE Usage](image)

In networks with composite PCE nodes as specified in section 5.1 of [RFC4655], PCE is implemented on several routers in the network, and the PCCs in the network can use the mechanism described in [RFC8231] to report the TE Policy information to the PCE nodes. An external component may also need to collect the TE Policy information from all the PCEs in the network to obtain a global view of the LSP state in the network.

In multi-area or multi-AS scenarios, each area or AS can have a child PCE to collect the TE Policies in its own domain, in addition, a parent PCE needs to collect TE Policy information from multiple child PCEs to obtain a global view of LSPs inside and across the domains involved.

In another network scenario, a centralized controller is used for service placement. Obtaining the TE Policy state information is quite important for making appropriate service placement decisions with the purpose to both meet the application’s requirements and utilize network resources efficiently.
The Network Management System (NMS) may need to provide global visibility of the TE Policies in the network as part of the network visualization function.

BGP has been extended to distribute link-state and traffic engineering information to external components [RFC7752]. Using the same protocol to collect Traffic Engineering Policy and state information is desirable for these external components since this avoids introducing multiple protocols for network information collection. This document describes a mechanism to distribute traffic engineering policy information (MPLS, SR, IPv4 and IPv6) to external components using BGP-LS.

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. Carrying TE Policy Information in BGP

TE Policy information is advertised in BGP UPDATE messages using the MP_REACH_NLRI and MP_UNREACH_NLRI attributes [RFC4760]. The "Link-State NLRI" defined in [RFC7752] is extended to carry the TE Policy information. BGP speakers that wish to exchange TE Policy information MUST use the BGP Multiprotocol Extensions Capability Code (1) to advertise the corresponding (AFI, SAFI) pair, as specified in [RFC4760]. New TLVs carried in the Link State Attribute defined in [RFC7752] are also defined in order to carry the attributes of a TE Policy in the subsequent sections.

The format of "Link-State NLRI" is defined in [RFC7752] as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            NLRI Type          |     Total NLRI Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                  Link-State NLRI (variable)                 //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

A new "NLRI Type" is defined for TE Policy Information as following:

- NLRI Type: TE Policy NLRI value 5.
The format of this new NLRI type is defined in Section 3 below.

3. TE Policy NLRI

This document defines the new TE Policy NLRI-Type and its format as shown in the following figure:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Protocol-ID |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Identifier                      |
|                        (64 bits)                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                Headend (Node Descriptors)                   //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                TE Policy Descriptors (variable)             //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Protocol-ID field specifies the component that owns the TE Policy state in the advertising node. The following new Protocol-IDs are defined and apply to the TE Policy NLRI:

```
+-------------+----------------------------------+
| Protocol-ID | NLRI information source protocol |
+-------------+----------------------------------+
|     8       |   RSVP-TE                        |
|     9       |   Segment Routing                |
```

- "Identifier" is an 8 octet value as defined in [RFC7752].

- "Headend" consists of a Local Node Descriptor (TLV 256) as defined in [RFC7752].

- "TE Policy Descriptors" consists of one or more of the TLVs listed as below:
The Local Node Descriptor TLV MUST include the following Node Descriptor TLVs:

- BGP Router-ID (TLV 516) [RFC9086], which contains a valid BGP Identifier of the local node.

- Autonomous System Number (TLV 512) [RFC7752], which contains the ASN or AS Confederation Identifier (ASN) [RFC5065], if confederations are used, of the local node.

The Local Node Descriptor TLV SHOULD include the following Node Descriptor TLVs:

- IPv4 Router-ID of Local Node (TLV 1028) [RFC7752], which contains the IPv4 TE Router-ID of the local node when one is provisioned.

- IPv6 Router-ID of Local Node (TLV 1029) [RFC7752], which contains the IPv6 TE Router-ID of the local node when one is provisioned.

The Local Node Descriptor TLV MAY include the following Node Descriptor TLVs:

- Member-ASN (TLV 517) [RFC9086], which contains the ASN of the confederation member (i.e., Member-AS Number), if BGP confederations are used, of the local node.

- Node Descriptors as defined in [RFC7752].

4. TE Policy Descriptors

This section defines the TE Policy Descriptors TLVs which are used to describe the TE Policy being advertised by using the new BGP-LS TE Policy NLRI type defined in Section 3.
4.1. Tunnel Identifier (Tunnel ID)

The Tunnel Identifier TLV contains the Tunnel ID defined in [RFC3209] and is used for RSVP-TE protocol TE Policies. It has the following format:

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Tunnel ID</th>
</tr>
</thead>
</table>

where:
- Type: 550
- Length: 2 octets.
- Tunnel ID: 2 octets as defined in [RFC3209].

4.2. LSP Identifier (LSP ID)

The LSP Identifier TLV contains the LSP ID defined in [RFC3209] and is used for RSVP-TE protocol TE Policies. It has the following format:

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>LSP ID</th>
</tr>
</thead>
</table>

where:
- Type: 551
- Length: 2 octets.
- LSP ID: 2 octets as defined in [RFC3209].
4.3. IPv4/IPv6 Tunnel Head-End Address

The IPv4/IPv6 Tunnel Head-End Address TLV contains the Tunnel Head-End Address defined in [RFC3209] and is used for RSVP-TE protocol TE Policies. It has following format:

```
 0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type              |          Length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//        IPv4/IPv6 Tunnel Head-End Address (variable)         //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- **Type**: 552
- **Length**: 4 or 16 octets.

When the IPv4/IPv6 Tunnel Head-end Address TLV contains an IPv4 address, its length is 4 (octets).

When the IPv4/IPv6 Tunnel Head-end Address TLV contains an IPv6 address, its length is 16 (octets).

4.4. IPv4/IPv6 Tunnel Tail-End Address

The IPv4/IPv6 Tunnel Tail-End Address TLV contains the Tunnel Tail-End Address defined in [RFC3209] and is used for RSVP-TE protocol TE Policies. It has following format:

```
 0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type              |          Length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//        IPv4/IPv6 Tunnel Tail-End Address (variable)         //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- **Type**: 553
- **Length**: 4 or 16 octets.

When the IPv4/IPv6 Tunnel Tail-end Address TLV contains an IPv4 address, its length is 4 (octets).
When the IPv4/IPv6 Tunnel Tail-end Address TLV contains an IPv6 address, its length is 16 (octets).

4.5. SR Policy Candidate Path Descriptor

The SR Policy Candidate Path Descriptor TLV identifies a Segment Routing Policy candidate path (CP) as defined in [I-D.ietf-spring-segment-routing-policy] and has the following format:

<table>
<thead>
<tr>
<th>0 1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3</td>
<td>4 5 6 7</td>
<td>8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
</tr>
<tr>
<td>+-------------------------------+-------------------------------+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Length</td>
<td></td>
</tr>
<tr>
<td>+-------------------------------+-------------------------------+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protocol-origin</td>
<td>Flags</td>
<td>RESERVED</td>
</tr>
<tr>
<td>+-------------------------------+-------------------------------+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Endpoint (4 or 16 octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-------------------------------+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Policy Color (4 octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-------------------------------+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Originator AS Number (4 octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-------------------------------+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Originator Address (4 or 16 octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-------------------------------+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discriminator (4 octets)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

where:

- **Type**: 554
- **Length**: variable (valid values are 24, 36 or 48 octets)
- **Protocol-Origin**: 1 octet field which identifies the protocol or component which is responsible for the instantiation of this path. Following protocol-origin codepoints are defined in this document.

<table>
<thead>
<tr>
<th>Code Point</th>
<th>Protocol Origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PCEP</td>
</tr>
<tr>
<td>2</td>
<td>BGP SR Policy</td>
</tr>
<tr>
<td>3</td>
<td>Local (via CLI, Yang model through NETCONF, gRPC, etc.)</td>
</tr>
</tbody>
</table>
4.6. Local MPLS Cross Connect

The Local MPLS Cross Connect TLV identifies a local MPLS state in the form of incoming label and interface followed by an outgoing label and interface. Outgoing interface may appear multiple times (for multicast states). It is used with Protocol ID set to "Static Configuration" value 5 as defined in [RFC7752].

The Local MPLS Cross Connect TLV has the following format:
where:

- Type: 555
- Length: variable.
- Incoming and Outgoing labels: 4 octets each.
- Sub-TLVs: following Sub-TLVs are defined:
  * Interface Sub-TLV
  * Forwarding Equivalent Class (FEC)

The Local MPLS Cross Connect TLV:

MUST have an incoming label.

MUST have an outgoing label.

MAY contain an Interface Sub-TLV having the I-flag set.

MUST contain at least one Interface Sub-TLV having the I-flag unset.

MAY contain multiple Interface Sub-TLV having the I-flag unset. This is the case of a multicast MPLS cross connect.

MAY contain a FEC Sub-TLV.

The following sub-TLVs are defined for the Local MPLS Cross Connect TLV:
These are defined in the following sub-sections.

4.6.1. MPLS Cross Connect Interface

The MPLS Cross Connect Interface sub-TLV is optional and contains the identifier of the interface (incoming or outgoing) in the form of an IPv4/IPv6 address and/or a local interface identifier.

The MPLS Cross Connect Interface sub-TLV has the following format:

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type              |          Length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Flags     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |          Local Interface Identifier (4 octets)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 //  Interface Address (4 or 16 octets)  //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

where:

- Type: 556
- Length: 9 or 21.
- Flags: 1 octet of flags defined as follows:

  0 1 2 3 4 5 6 7
  ++++++++---
  |I|       |
  ++++++++---

where:

* I-Flag is the Interface flag. When set, the Interface Sub-TLV describes an incoming interface. If the I-flag is not set, then the Interface Sub-TLV describes an outgoing interface.

- Local Interface Identifier: a 4 octet identifier.
- Interface address: a 4 octet IPv4 address or a 16 octet IPv6 address.

### 4.6.2. MPLS Cross Connect FEC

The MPLS Cross Connect FEC sub-TLV is optional and contains the FEC associated to the incoming label.

The MPLS Cross Connect FEC sub-TLV has the following format:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type              |          Length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Flags       |  Masklength   |   Prefix (variable)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                     Prefix (variable)                       //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                     Prefix (variable)                       //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 557
- Length: variable.
- Flags: 1 octet of flags defined as follows:

```
0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|4|          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

* 4-Flag is the IPv4 flag. When set, the FEC Sub-TLV describes an IPv4 FEC. If the 4-flag is not set, then the FEC Sub-TLV describes an IPv6 FEC.

- Mask Length: 1 octet of prefix length.
5. MPLS-TE Policy State TLV

A new TLV called "MPLS-TE Policy State TLV", is used to describe the characteristics of the MPLS-TE Policy and it is carried in the optional non-transitive BGP Attribute "LINK_STATE Attribute" defined in [RFC7752]. These MPLS-TE Policy characteristics include the characteristics and attributes of the policy, its dataplane, explicit path, Quality of Service (QoS) parameters, route information, the protection mechanisms, etc.

The MPLS-TE Policy State TLV has the following format:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Object-origin | Address Family|            RESERVED           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//        MPLS-TE Policy State Objects (variable)              //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

where:

MPLS-TE Policy State TLV

- Type: 1200

- Length: the total length of the MPLS-TE Policy State TLV not including Type and Length fields.

- Object-origin: identifies the component (or protocol) from which the contained object originated. This allows for objects defined in different components to be collected while avoiding the possible codepoint collisions among these components. Following object-origin codepoints are defined in this document.
o Address Family: describes the address family used to setup the MPLS-TE policy. The following address family values are defined in this document:

```
+----------+------------------+
|  Code    |    Dataplane     |
|  Point   |                  |
+----------+------------------+
|    1     | MPLS-IPv4        |
|    2     | MPLS-IPv6        |
```

o RESERVED: 16-bit field. SHOULD be set to 0 on transmission and MUST be ignored on receipt.

o TE Policy State Objects: Rather than replicating all these objects in this document, the semantics and encodings of the objects as defined in RSVP-TE and PCEP are reused.

The state information is carried in the "MPLS-TE Policy State Objects" with the following format as described in the sub-sections below.

5.1. RSVP Objects

RSVP-TE objects are encoded in the "MPLS-TE Policy State Objects" field of the MPLS-TE Policy State TLV and consists of MPLS TE LSP objects defined in RSVP-TE [RFC3209] [RFC3473]. Rather than replicating all MPLS TE LSP related objects in this document, the semantics and encodings of the MPLS TE LSP objects are re-used. These MPLS TE LSP objects are carried in the MPLS-TE Policy State TLV.

When carrying RSVP-TE objects, the "Object-Origin" field is set to "RSVP-TE".

The following RSVP-TE Objects are defined:

- SENDER_TSPEC and FLOW_SPEC [RFC2205]
For the MPLS TE LSP Objects listed above, the corresponding sub-objects are also applicable to this mechanism. Note that this list is not exhaustive, other MPLS TE LSP objects which reflect specific characteristics of the MPLS TE LSP can also be carried in the LSP state TLV.

5.2. PCEP Objects

PCEP objects are encoded in the "MPLS-TE Policy State Objects" field of the MPLS-TE Policy State TLV and consists of PCEP objects defined in [RFC5440]. Rather than replicating all MPLS TE LSP related objects in this document, the semantics and encodings of the MPLS TE LSP objects are re-used. These MPLS TE LSP objects are carried in the MPLS-TE Policy State TLV.

When carrying PCEP objects, the "Object-Origin" field is set to "PCEP".
The following PCEP Objects are defined:

- METRIC Object [RFC5440]
- BANDWIDTH Object [RFC5440]

For the MPLS TE LSP Objects listed above, the corresponding sub-objects are also applicable to this mechanism. Note that this list is not exhaustive, other MPLS TE LSP objects which reflect specific characteristics of the MPLS TE LSP can also be carried in the TE Policy State TLV.

6. SR Policy State TLVs

Segment Routing Policy (SR Policy) architecture is specified in [I-D.ietf-spring-segment-routing-policy]. A SR Policy can comprise of one or more candidate paths (CP) of which at a given time one and only one may be active (i.e. installed in forwarding and usable for steering of traffic). Each CP in turn may have one or more SID-List of which one or more may be active; when multiple are active then traffic is load balanced over them.

This section defines the various TLVs which enable the headend to report the state of an SR Policy, its CP(s), SID-List(s) and their status. These TLVs are carried in the optional non-transitive BGP Attribute "LINK_STATE Attribute" defined in [RFC7752] and enable the same consistent form of reporting for SR Policy state irrespective of the Protocol-Origin used to provision the policy. Detailed procedure is described in Section 7.

6.1. SR Binding SID

The SR Binding SID (BSID) is an optional TLV that provides the BSID and its attributes for the SR Policy CP. The TLV MAY also optionally contain the Specified BSID value for reporting as described in section 6.2.3 of [I-D.ietf-spring-segment-routing-policy].

The TLV has the following format:
Internet-Draft  TE Policy State Distribution using BGP-LS     April 2022

0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           BSID Flags          |            RESERVED           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Binding SID (4 or 16 octets)               //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Specified Binding SID (4 or 16 octets)         //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

where:

- Type: 1201
- Length: variable (valid values are 12 or 36 octets)
- BSID Flags: 2 octet field that indicates attribute and status of the Binding SID (BSID) associated with this CP. The following bit positions are defined and the semantics are described in detail in [I-D.ietf-spring-segment-routing-policy]. Other bits SHOULD be cleared by originator and MUST be ignored by receiver.

0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|D|B|U|L|F|                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

where:

- D-Flag : Indicates the dataplane for the BSIDs and if they are 16 octet SRv6 SID when set and are 4 octet SR/MPLS label value when clear.
- B-Flag : Indicates the allocation of the value in the BSID field when set and indicates that BSID is not allocated when clear.
- U-Flag : Indicates the specified BSID value is unavailable when set.
- L-Flag : Indicates the BSID value is from the Segment Routing Local Block (SRLB) of the headend node when set and is from the local dynamic label pool when clear.
* F-Flag : Indicates the BSID value is one allocated from dynamic label pool due to fallback (e.g. when specified BSID is unavailable) when set.

- RESERVED: 2 octets. SHOULD be set to 0 by originator and MUST be ignored by receiver.
- Binding SID: It indicates the operational or allocated BSID value for the CP based on the status flags.
- Specified BSID: It is used to report the explicitly specified BSID value regardless of whether it is successfully allocated or not. The field is set to value 0 when BSID has not been specified for the CP.

The BSID fields above are 4 octet carrying the MPLS Label or 16 octets carrying the SRv6 SID based on the BSID D-flag. When carrying the MPLS Label, as shown in the figure below, the TC, S and TTL (total of 12 bits) are RESERVED and SHOULD be set to 0 by originator and MUST be ignored by the receiver.

```
+---------------------------------+  +---------------------------------+
|                                   |  |                                   |
|                                   |  |                                   |
|                                   |  |                                   |
|                                   |  |                                   |
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1|
|--------------------------------------|  |--------------------------------------|
```

In the case of an SRv6, the Binding SID sub-TLV does not have the ability to signal the SRv6 Endpoint Behavior [RFC8986] or the structure of the SID. It is RECOMMENDED that the SRv6 Binding SID TLV defined in Section 6.2, that enables the specification of the SRv6 Endpoint Behavior, be used for signaling of an SRv6 Binding SID for an SR Policy candidate path.

6.2. SRv6 Binding SID

The SRv6 Binding SID (BSID) is an optional TLV that provides the SRv6 BSID and its attributes for the SR Policy CP. The TLV MAY also optionally contain the Specified SRv6 BSID value for reporting as described in section 6.2.3 of [I-D.ietf-spring-segment-routing-policy].

The TLV has the following format:
where:

- **Type**: TBD
- **Length**: variable
- **BSID Flags**: 2 octet field that indicates attribute and status of the Binding SID (BSID) associated with this CP. The following bit positions are defined and the semantics are described in detail in [I-D.ietf-spring-segment-routing-policy]. Other bits SHOULD be cleared by originator and MUST be ignored by receiver.

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           BSID Flags          |            RESERVED           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Binding SID (16 octets)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Specified Binding SID (16 octets)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
// Sub-TLVs (variable)                                         //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
where:
- B-Flag : Indicates the allocation of the value in the BSID field when set and indicates that BSID is not allocated when clear.
- U-Flag : Indicates the specified BSID value is unavailable when set.
- F-Flag : Indicates the BSID value is one allocated dynamically due to fallback (e.g. when specified BSID is unavailable) when set.
- RESERVED: 2 octets. SHOULD be set to 0 by originator and MUST be ignored by receiver.
o Binding SID: It indicates the operational or allocated BSID value for the CP based on the status flags.

o Specified BSID: It is used to report the explicitly specified BSID value regardless of whether it is successfully allocated or not. The field is set to value 0 when BSID has not been specified for the CP.

o Sub-TLVs: variable and contains any other optional attributes associated with the SRv6 BSID.

The SRv6 Endpoint Behavior TLV (1250) and the SRv6 SID Structure TLV (1252) defined in [I-D.ietf-idr-bgpls-srv6-ext] are used as sub-TLVs of the SRv6 Binding SID TLV to optionally indicate the SRv6 Endpoint behavior and SID structure for the Binding SID value in the TLV.

### 6.3. SR Candidate Path State

The SR Candidate Path (CP) State TLV provides the operational status and attributes of the SR Policy at the CP level. The TLV has the following format:

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +---------------+---------------+---------------+---------------+
  | Type           | Length         |
  +---------------+---------------+---------------+---------------+
  | Priority       | RESERVED      | Flags         |
  +---------------+---------------+---------------+---------------+
  | Preference (4 octets) |
  +---------------+---------------+---------------+---------------+
```

where:

- **Type**: 1202
- **Length**: 8 octets
- **Priority**: 1 octet value which indicates the priority of the CP. Refer Section 2.12 of [I-D.ietf-spring-segment-routing-policy].
- **RESERVED**: 1 octet. SHOULD be set to 0 by originator and MUST be ignored by receiver.
- **Flags**: 2 octet field that indicates attribute and status of the CP. The following bit positions are defined and the semantics are described in detail in [I-D.ietf-spring-segment-routing-policy].
Other bits SHOULD be cleared by originator and MUST be ignored by receiver.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|S|A|B|E|V|O|D|C|I|T|           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

* S-Flag : Indicates the CP is in administrative shut state when set

* A-Flag : Indicates the CP is the active path (i.e. one provisioned in the forwarding plane) for the SR Policy when set

* B-Flag : Indicates the CP is the backup path (i.e. one identified for path protection of the active path) for the SR Policy when set

* E-Flag : Indicates that the CP has been evaluated for validity (e.g. headend may evaluate CPs based on their preferences) when set

* V-Flag : Indicates the CP has at least one valid SID-List when set. When the E-Flag is clear (i.e. the CP has not been evaluated), then this flag MUST be set to 0 by the originator and ignored by the receiver.

* O-Flag : Indicates the CP was instantiated by the headend due to an on-demand-nexthop trigger based on local template when set. Refer Section 8.5 of [I-D.ietf-spring-segment-routing-policy].

* D-Flag : Indicates the CP was delegated for computation to a PCE/controller when set

* C-Flag : Indicates the CP was provisioned by a PCE/controller when set

* I-Flag : Indicates the CP will perform the "drop upon invalid" behavior when no other active path is available for this SR Policy and this path is the one with best preference amongst the available CPs. Refer Section 8.2 of [I-D.ietf-spring-segment-routing-policy].
* T-Flag : Indicates the CP has been marked as eligible for use as Transit Policy on the headend when set. Refer Section 8.3 of [I-D.ietf-spring-segment-routing-policy].

- Preference : 4 octet value which indicates the preference of the CP. Refer Section 2.7 of [I-D.ietf-spring-segment-routing-policy].

6.4. SR Policy Name

The SR Policy Name TLV is an optional TLV that is used to carry the symbolic name associated with the SR Policy. The TLV has the following format:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Type            |              Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   SR Policy Name (variable)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: TBD
- Length: variable
- SR Policy Name: Symbolic name for the SR Policy without a NULL terminator as specified in section 2.6 of [I-D.ietf-spring-segment-routing-policy]. It is RECOMMENDED that the size of the symbolic name be limited to 255 bytes. Implementations MAY choose to truncate long names to 255 bytes when signaling via BGP-LS.

6.5. SR Candidate Path Name

The SR Candidate Path Name TLV is an optional TLV that is used to carry the symbolic name associated with the candidate path. The TLV has the following format:
o Type: 1203

o Length: variable

o Candidate Path Name: Symbolic name for the SR Policy candidate path without a NULL terminator as specified in section 2.6 of [I-D.ietf-spring-segment-routing-policy]. It is RECOMMENDED that the size of the symbolic name be limited to 255 bytes. Implementations MAY choose to truncate long names to 255 bytes when signaling via BGP-LS.

6.6. SR Candidate Path Constraints

The SR Candidate Path Constraints TLV is an optional TLV that is used to report the constraints associated with the candidate path. The constraints are generally applied to a dynamic candidate path which is computed by the headend. The constraints may also be applied to an explicit path where the headend is expected to validate that the path expresses satisfies the specified constraints and the path is to be invalidated by the headend when the constraints are no longer met (e.g. due to topology changes).

The TLV has the following format:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Type            |              Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Flags            |           RESERVED            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             MTID              |   Algorithm   |    RESERVED   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   sub-TLVs (variable)                                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

where:
```
o Type: 1204

o Length: variable

o Flags: 2 octet field that indicates the constraints that are being applied to the CP. The following bit positions are defined and the other bits SHOULD be cleared by originator and MUST be ignored by receiver.

```
 0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|D|P|U|A|T|                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

where:

* D-Flag : Indicates that the CP needs to use SRv6 dataplane when set and SR/MPLS dataplane when clear

* P-Flag : Indicates that the CP needs to use only protected SIDs when set

* U-Flag : Indicates that the CP needs to use only unprotected SIDs when set

* A-Flag : Indicates that the CP needs to use the SIDs belonging to the specified SR Algorithm only when set

* T-Flag: Indicates that the CP needs to use the SIDs belonging to the specified topology only when set

o RESERVED: 2 octet. SHOULD be set to 0 by originator and MUST be ignored by receiver.

o MTID : Indicates the multi-topology identifier of the IGP topology that is preferred to be used when the path is setup. When the T-flag is set then the path is strictly using the specified topology SIDs only.

o Algorithm : Indicates the algorithm that is preferred to be used when the path is setup. When the A-flag is set then the path is strictly using the specified algorithm SIDs only.

o RESERVED: 1 octet. SHOULD be set to 0 by originator and MUST be ignored by receiver.
sub-TLVs: optional sub-TLVs MAY be included in this TLV to describe other constraints.

The following constraint sub-TLVs are defined for the SR CP Constraints TLV.

### 6.6.1. SR Affinity Constraint

The SR Affinity Constraint sub-TLV is an optional sub-TLV that is used to carry the affinity constraints [RFC2702] associated with the candidate path. The affinity is expressed in terms of Extended Admin Group (EAG) as defined in [RFC7308]. The TLV has the following format:

```
  0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Type            |              Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Excl-Any-Size | Incl-Any-Size | Incl-All-Size |    RESERVED   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Exclude-Any EAG (optional, variable)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Include-Any EAG (optional, variable)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Include-All EAG (optional, variable)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- **Type:** 1208

- **Length:** variable, dependent on the size of the Extended Admin Group. MUST be a multiple of 4 octets.

- **Exclude-Any-Size:** one octet to indicate the size of Exclude-Any EAG bitmask size in multiples of 4 octets. (e.g. value 0 indicates the Exclude-Any EAG field is skipped, value 1 indicates that 4 octets of Exclude-Any EAG is included)

- **Include-Any-Size:** one octet to indicate the size of Include-Any EAG bitmask size in multiples of 4 octets. (e.g. value 0 indicates the Include-Any EAG field is skipped, value 1 indicates that 4 octets of Include-Any EAG is included)

- **Include-All-Size:** one octet to indicate the size of Include-All EAG bitmask size in multiples of 4 octets. (e.g. value 0
indicates the Include-All EAG field is skipped, value 1 indicates that 4 octets of Include-All EAG is included

- RESERVED: 1 octet. SHOULD be set to 0 by originator and MUST be ignored by receiver.
- Exclude-Any EAG: the bitmask used to represent the affinities to be excluded from the path.
- Include-Any EAG: the bitmask used to represent the affinities to be included in the path.
- Include-All EAG: the bitmask used to represent the all affinities to be included in the path.

6.6.2. SR SRLG Constraint

The SR SRLG Constraint sub-TLV is an optional sub-TLV that is used to carry the Shared Risk Link Group (SRLG) values [RFC4202] that are to be excluded from the candidate path. The TLV has the following format:

```
+-----------------+-----------------+-----------------+-----------------+
|               Type            |              Length           |
+-----------------+-----------------+-----------------+-----------------+
| SRLG Values (variable, multiples of 4 octets)        |
+-----------------+-----------------+-----------------+-----------------+
```

where:

- Type: 1209
- Length: variable, dependent on the number of SRLGs encoded. MUST be a multiple of 4 octets.
- SRLG Values: One or more SRLG values (each of 4 octets).

6.6.3. SR Bandwidth Constraint

The SR Bandwidth Constraint sub-TLV is an optional sub-TLV that is used to indicate the desired bandwidth availability that needs to be ensured for the candidate path. The TLV has the following format:
where:

- Type: 1210
- Length: 4 octets
- Bandwidth: 4 octets which specify the desired bandwidth in units of bytes per second in IEEE floating point format.

### 6.6.4. SR Disjoint Group Constraint

The SR Disjoint Group Constraint sub-TLV is an optional sub-TLV that is used to carry the disjointness constraint associated with the candidate path. The disjointness between two SR Policy Candidate Paths is expressed by associating them with the same disjoint group identifier and then specifying the type of disjointness required between their paths. The computation is expected to achieve the highest level of disjointness requested and when that is not possible then fallback to a lesser level progressively based on the levels indicated.

The TLV has the following format:

![TLV Format](image)

where:

- Type: 1211
- Length: 8 octets
Request Flags : one octet to indicate the level of disjointness requested as specified in the form of flags. The following flags are defined and the other bits SHOULD be cleared by originator and MUST be ignored by receiver.

```
0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+
|S|N|L|F|I|     |
+-+-+-+-+-+-+-+
```

where:

* S-Flag : Indicates that SRLG disjointness is requested
* N-Flag : Indicates that node disjointness is requested when
* L-Flag : Indicates that link disjointness is requested when
* F-Flag : Indicates that the computation may fallback to a lower level of disjointness amongst the ones requested when all cannot be achieved
* I-Flag : Indicates that the computation may fallback to the default best path (e.g. IGP path) in case of none of the desired disjointness can be achieved.

Status Flags : one octet to indicate the level of disjointness that has been achieved by the computation as specified in the form of flags. The following flags are defined and the other bits SHOULD be cleared by originator and MUST be ignored by receiver.

```
0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+
|S|N|L|F|I|X|   |
+-+-+-+-+-+-+-+
```

where:

* S-Flag : Indicates that SRLG disjointness is achieved
* N-Flag : Indicates that node disjointness is achieved
* L-Flag : Indicates that link disjointness is achieved
* F-Flag : Indicates that the computation has fallen back to a lower level of disjointness that requested.
* I-Flag: Indicates that the computation has fallen back to the best path (e.g., IGP path) and disjointness has not been achieved

* X-Flag: Indicates that the disjointness constraint could not be achieved and hence path has been invalidated

o RESERVED: 2 octets. SHOULD be set to 0 by originator and MUST be ignored by receiver.

o Disjointness Group Identifier: 4 octet value that is the group identifier for a set of disjoint paths

6.7. SR Segment List

The SR Segment List TLV is used to report the SID-List(s) of a candidate path. The TLV has following format:

```
<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flags</td>
<td>RESERVED</td>
</tr>
<tr>
<td>MTID</td>
<td>Algorithm</td>
</tr>
<tr>
<td>Weight (4 octets)</td>
<td></td>
</tr>
</tbody>
</table>
```

where:

o Type: 1205

o Length: variable

o Flags: 2 octet field that indicates attribute and status of the SID-List. The following bit positions are defined and the semantics are described in detail in [I-D.ietf-spring-segment-routing-policy]. Other bits SHOULD be cleared by originator and MUST be ignored by receiver.
0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|D|E|C|V|R|F|A|T|M|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

where:

* D-Flag : Indicates the SID-List is comprised of SRv6 SIDs when set and indicates it is comprised of SR/MPLS labels when clear.

* E-Flag : Indicates that SID-List is an explicit path when set and indicates dynamic path when clear.

* C-Flag : Indicates that SID-List has been computed for a dynamic path when set. It is always reported as set for explicit paths.

* V-Flag : Indicates the SID-List has passed verification or its verification was not required when set and failed verification when clear.

* R-Flag : Indicates that the first Segment has been resolved when set and failed resolution when clear.

* F-Flag : Indicates that the computation for the dynamic path failed when set and succeeded (or not required in case of explicit path) when clear.

* A-Flag : Indicates that all the SIDs in the SID-List belong to the specified algorithm when set.

* T-Flag : Indicates that all the SIDs in the SID-List belong to the specified topology (identified by the multi-topology ID) when set.

* M-Flag : Indicates that the SID-list has been removed from the forwarding plane due to fault detection by a monitoring mechanism (e.g. BFD) when set and indicates no fault detected or monitoring is not being done when clear.

  o RESERVED: 2 octet. SHOULD be set to 0 by originator and MUST be ignored by receiver.

  o MTID : 2 octet that indicates the multi-topology identifier of the IGP topology to be used when the T-flag is set.
o Algorithm: 1 octet that indicates the algorithm of the SIDs used in the SID-List when the A-flag is set.

o RESERVED: 1 octet. SHOULD be set to 0 by originator and MUST be ignored by receiver.

o Weight: 4 octet field that indicates the weight associated with the SID-List for weighted load-balancing. Refer Section 2.2 and 2.11 of [I-D.ietf-spring-segment-routing-policy].

o Sub-TLVs: variable and contains the ordered set of Segments and any other optional attributes associated with the specific SID-List.

The SR Segment sub-TLV (defined in Section 6.8) MUST be included as an ordered set of sub-TLVs within the SR Segment List TLV when the SID-List is not empty. A SID-List may be empty in certain cases (e.g. for a dynamic path) where the headend has not yet performed the computation and hence not derived the segments required for the path; in such cases, the SR Segment List TLV SHOULD NOT include any SR Segment sub-TLVs.

### 6.8. SR Segment

The SR Segment sub-TLV describes a single segment in a SID-List. One or more instances of this sub-TLV in an ordered manner constitute a SID-List for a SR Policy candidate path. It is a sub-TLV of the SR Segment List TLV and has following format:

```
+-------------------+-------------------+-------------------+-------------------+
|              Type |             Length |
+-------------------+-------------------+-------------------+-------------------+
| Segment Type  |   RESERVED    |             Flags |
+-------------------+-------------------+-------------------+-------------------+
|                   SID (4 or 16 octets)                       |
+-------------------+-------------------+-------------------+-------------------+-------------------+
//               Segment Descriptor (variable)               |
+-------------------+-------------------+-------------------+-------------------+-------------------+
//                 Sub-TLVs (variable)                         |
+-------------------+-------------------+-------------------+-------------------+-------------------+
```

where:

- Type: 1206
- Length: variable
o Segment Type : 1 octet which indicates the type of segment (refer Section 6.8.1 for details)

o RESERVED: 1 octet. SHOULD be set to 0 by originator and MUST be ignored by receiver.

o Flags: 2 octet field that indicates attribute and status of the Segment and its SID. The following bit positions are defined and the semantics are described in detail in [I-D.ietf-spring-segment-routing-policy]. Other bits SHOULD be cleared by originator and MUST be ignored by receiver.

```
0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|S|E|V|R|A|                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

* S-Flag : Indicates the presence of SID value in the SID field when set and that no value is indicated when clear.

* E-Flag : Indicates the SID value is explicitly provisioned value (locally on headend or via controller/PCE) when set and is a dynamically resolved value by headend when clear

* V-Flag : Indicates the SID has passed verification or did not require verification when set and failed verification when clear.

* R-Flag : Indicates the SID has been resolved or did not require resolution (e.g. because it is not the first SID) when set and failed resolution when clear.

* A-Flag : Indicates that the Algorithm indicated in the Segment descriptor is valid when set. When clear, it indicates that the headend is unable to determine the algorithm of the SID.

o SID : 4 octet carrying the MPLS Label or 16 octets carrying the SRv6 SID based on the Segment Type. When carrying the MPLS Label, as shown in the figure below, the TC, S and TTL (total of 12 bits) are RESERVED and SHOULD be set to 0 by originator and MUST be ignored by the receiver.
6.8.1. Segment Descriptors

[I-D.ietf-spring-segment-routing-policy] section 4 defines multiple types of segments and their description. This section defines the encoding of the Segment Descriptors for each of those Segment types to be used in the Segment sub-TLV describes previously in Section 6.8.

The following types are currently defined:

- Segment Descriptor: variable size Segment descriptor based on the type of segment (refer Section 6.8.1 for details)
- Sub-Sub-TLVs: variable and contains any other optional attributes associated with the specific segment.

The SRv6 Endpoint Behavior TLV (1250) and the SRv6 SID Structure TLV (1252) defined in [I-D.ietf-idr-bgppls-srv6-ext] are used as sub-sub-TLVs of the SR Segment sub-TLV to optionally indicate the SRv6 Endpoint behavior and SID structure when advertising the SRv6 specific segment types.
<table>
<thead>
<tr>
<th>Type</th>
<th>Segment Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Invalid</td>
</tr>
<tr>
<td>1</td>
<td>SR-MPLS Label</td>
</tr>
<tr>
<td>2</td>
<td>SRv6 SID as IPv6 address</td>
</tr>
<tr>
<td>3</td>
<td>SR-MPLS Prefix SID as IPv4 Node Address</td>
</tr>
<tr>
<td>4</td>
<td>SR-MPLS Prefix SID as IPv6 Node Global Address</td>
</tr>
<tr>
<td>5</td>
<td>SR-MPLS Adjacency SID as IPv4 Node Address &amp; Local Interface ID</td>
</tr>
<tr>
<td>6</td>
<td>SR-MPLS Adjacency SID as IPv4 Local &amp; Remote Interface Addresses</td>
</tr>
<tr>
<td>7</td>
<td>SR-MPLS Adjacency SID as pair of IPv6 Global Address &amp; Interface ID for Local &amp; Remote nodes</td>
</tr>
<tr>
<td>8</td>
<td>SR-MPLS Adjacency SID as pair of IPv6 Global Addresses for the Local &amp; Remote Interface</td>
</tr>
<tr>
<td>9</td>
<td>SRv6 END SID as IPv6 Node Global Address</td>
</tr>
<tr>
<td>10</td>
<td>SRv6 END.X SID as pair of IPv6 Global Address &amp; Interface ID for Local &amp; Remote nodes</td>
</tr>
<tr>
<td>11</td>
<td>SRv6 END.X SID as pair of IPv6 Global Addresses for the Local &amp; Remote Interface</td>
</tr>
</tbody>
</table>

### 6.8.1.1. Type 1: SR-MPLS Label

The Segment is SR-MPLS type and is specified simply as the label. The format of its Segment Descriptor is as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------+-+-+++
| Algorithm |
+-------+-+-+++
```

Where:

- **Algorithm**: 1 octet value that indicates the algorithm used for picking the SID. This is valid only when the A-flag has been set in the Segment TLV.

### 6.8.1.2. Type 2: SRv6 SID

The Segment is SRv6 type and is specified simply as the SRv6 SID address. The format of its Segment Descriptor is as follows:
Where:

- Algorithm: 1 octet value that indicates the algorithm used for picking the SID. This is valid only when the A-flag has been set in the Segment TLV.

6.8.1.3. Type 3: SR-MPLS Prefix SID for IPv4

The Segment is SR-MPLS Prefix SID type and is specified as an IPv4 node address. The format of its Segment Descriptor is as follows:

```
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +--------------------------------------------------+
  | Algorithm | IPv4 Node Address (4 octets) |
  +--------------------------------------------------+
```

Where:

- Algorithm: 1 octet value that indicates the algorithm used for picking the SID
- IPv4 Node Address: 4 octet value which carries the IPv4 address associated with the node

6.8.1.4. Type 4: SR-MPLS Prefix SID for IPv6

The Segment is SR-MPLS Prefix SID type and is specified as an IPv6 global address. The format of its Segment Descriptor is as follows:

```
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +--------------------------------------------------+
  | Algorithm | IPv6 Node Global Address (16 octets) |
  +--------------------------------------------------+
```

Where:
o Algorithm: 1 octet value that indicates the algorithm used for picking the SID

o IPv6 Node Global Address: 16 octet value which carries the IPv6 global address associated with the node

6.8.1.5. Type 5: SR-MPLS Adjacency SID for IPv4 with Interface ID

The Segment is SR-MPLS Adjacency SID type and is specified as an IPv4 node address along with the local interface ID on that node. The format of its Segment Descriptor is as follows:

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    IPv4 Node Address (4 octets)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Local Interface ID (4 octets)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

o IPv4 Node Address: 4 octet value which carries the IPv4 address associated with the node

o Local Interface ID : 4 octet value which carries the local interface ID of the node identified by the Node Address

6.8.1.6. Type 6: SR-MPLS Adjacency SID for IPv4 with Interface Address

The Segment is SR-MPLS Adjacency SID type and is specified as a pair of IPv4 local and remote addresses. The format of its Segment Descriptor is as follows:

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    IPv4 Local Address (4 octets)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                IPv4 Remote Address (4 octets)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

o IPv4 Local Address: 4 octet value which carries the local IPv4 address associated with the node
IPv4 Remote Address: 4 octet value which carries the remote IPv4 address associated with the node’s neighbor. This is optional and MAY be set to 0 when not used (e.g. when identifying point-to-point links).

6.8.1.7. Type 7: SR-MPLS Adjacency SID for IPv6 with interface ID

The Segment is SR-MPLS Adjacency SID type and is specified as a pair of IPv6 global address and interface ID for local and remote nodes. The format of its Segment Descriptor is as follows:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IPv6 Local Node Global Address (16 octets) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Local Node Interface ID (4 octets) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IPv6 Remote Node Global Address (16 octets) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Remote Node Interface ID (4 octets) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

- IPv6 Local Node Global Address: 16 octet value which carries the IPv6 global address associated with the local node.
- Local Node Interface ID: 4 octet value which carries the interface ID of the local node identified by the Local Node Address.
- IPv6 Remote Node Global Address: 16 octet value which carries the IPv6 global address associated with the remote node. This is optional and MAY be set to 0 when not used (e.g. when identifying point-to-point links).
- Remote Node Interface ID: 4 octet value which carries the interface ID of the remote node identified by the Remote Node Address. This is optional and MAY be set to 0 when not used (e.g. when identifying point-to-point links).

6.8.1.8. Type 8: SR-MPLS Adjacency SID for IPv6 with interface address

The Segment is SR-MPLS Adjacency SID type and is specified as a pair of IPv6 Global addresses for local and remote interface addresses. The format of its Segment Descriptor is as follows:
Where:

- **IPv6 Local Address**: 16 octet value which carries the local IPv6 address associated with the node
- **IPv6 Remote Address**: 16 octet value which carries the remote IPv6 address associated with the node’s neighbor

6.8.1.9. Type 9: SRv6 END SID as IPv6 Node Address

The Segment is SRv6 END SID type and is specified as an IPv6 global address. The format of its Segment Descriptor is as follows:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Algorithm |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| IPv6 Node Global Address (16 octets) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Where:

- **Algorithm**: 1 octet value that indicates the algorithm used for picking the SID
- **IPv6 Node Global Address**: 16 octet value which carries the IPv6 global address associated with the node

6.8.1.10. Type 10: SRv6 END.X SID as interface ID

The Segment is SRv6 END.X SID type and is specified as a pair of IPv6 global address and interface ID for local and remote nodes. The format of its Segment Descriptor is as follows:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Algorithm  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| IPv6 Node Global Address (16 octets) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```
Where:

- IPv6 Local Node Global Address: 16 octet value which carries the IPv6 global address associated with the local node.

- Local Node Interface ID: 4 octet value which carries the interface ID of the local node identified by the Local Node Address.

- IPv6 Remote Node Global Address: 16 octet value which carries the IPv6 global address associated with the remote node. This is optional and MAY be set to 0 when not used (e.g. when identifying point-to-point links).

- Remote Node Interface ID: 4 octet value which carries the interface ID of the remote node identified by the Remote Node Address. This is optional and MAY be set to 0 when not used (e.g. when identifying point-to-point links).

6.8.1.11. Type 11: SRv6 END.X SID as interface address

The Segment is SRv6 END.X SID type and is specified as a pair of IPv6 Global addresses for local and remote interface addresses. The format of its Segment Descriptor is as follows:

```
0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IPv6 Local Node Global Address (16 octets) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Local Node Interface ID (4 octets) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IPv6 Remote Node Global Address (16 octets) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Remote Node Interface ID (4 octets) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:
6.9. SR Segment List Metric

The SR Segment List Metric sub-TLV describes the metric used for computation of the SID-List. It is used to report the type of metric used in the computation of a dynamic path either on the headend or when the path computation is delegated to a PCE/controller. When the path computation is done on the headend, it is also used to report the calculated metric for the path.

It is a sub-TLV of the SR Segment List TLV and has following format:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Metric Type  |      Flags    |          RESERVED             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Metric Margin                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Metric Bound                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Metric Value                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- **Type**: 1207
- **Length**: 16 octets
- **Metric Type**: 1 octet field which identifies the type of metric used for path computation. Following metric type codepoints are defined in this document.

<table>
<thead>
<tr>
<th>Code Point</th>
<th>Metric Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>IGP Metric</td>
</tr>
<tr>
<td>1</td>
<td>Min Unidirectional Link Delay [RFC7471]</td>
</tr>
<tr>
<td>2</td>
<td>TE Metric [RFC3630]</td>
</tr>
</tbody>
</table>
o Flags: 1 octet field that indicates the validity of the metric fields and their semantics. The following bit positions are defined and the other bits SHOULD be cleared by originator and MUST be ignored by receiver.

```
0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|M|A|B|V|       |
+-+-+-+-+-+-+-+-+
```

where:

* M-Flag : Indicates that the metric margin allowed for path computation is specified when set
* A-Flag : Indicates that the metric margin is specified as an absolute value when set and is expressed as a percentage of the minimum metric when clear.
* B-Flag : Indicates that the metric bound allowed for the path is specified when set.
* V-Flag : Indicates that the metric value computed is being reported when set.

o RESERVED: 2 octets. SHOULD be set to 0 by originator and MUST be ignored by receiver.

o Metric Margin : 4 octets which indicate the metric margin value when M-flag is set. The metric margin is specified as either an absolute value or as a percentage of the minimum computed path metric based on the A-flag. The metric margin loosens the criteria for minimum metric path calculation up to the specified metric to accommodate for other factors such as bandwidth availability, minimal SID stack depth and maximizing of ECMP for the SR path computed.

o Metric Bound : 4 octets which indicate the maximum metric value that is allowed when B-flag is set. If the computed path metric crosses the specified bound value then the path is considered as invalid.

o Metric Value : 4 octets which indicate the metric value of the computed path when V-flag is set. This value is available and reported when the computation is successful and a valid path is available.
7. Procedures

The BGP-LS advertisements for the TE Policy NLRI are originated by the headend node for the TE Policies that are instantiated on its local node.

For MPLS TE LSPs signaled via RSVP-TE, the NLRI descriptor TLVs as specified in Section 4.1, Section 4.2, Section 4.3 and Section 4.4 are used. Then the TE LSP state is encoded in the BGP-LS Attribute field as MPLS-TE Policy State TLV as described in Section 5. The RSVP-TE objects that reflect the state of the LSP are included as defined in Section 5.1. When the TE LSP is setup with the help of PCEP signaling then another MPLS-TE Policy State TLV SHOULD be used to encode the related PCEP objects corresponding to the LSP as defined in Section 5.2.

For SR Policies, the NLRI descriptor TLV as specified in Section 4.5 is used. An SR Policy candidate path (CP) may be instantiated on the headend node via a local configuration, PCEP or BGP SR Policy signaling and this is indicated via the SR Protocol Origin. Then the SR Policy Candidate Path’s attribute and state is encoded in the BGP-LS Attribute field as SR Policy State TLVs and sub-TLVs as described in Section 6. The SR Candidate Path State TLV as defined in Section 6.3 is included to report the state of the CP. The SR BSID TLV as defined in Section 6.1 or Section 6.2 is included to report the BSID of the CP when one is either specified or allocated by the headend. The constraints for the SR Policy Candidate Path are reported using the SR Candidate Path Constraints TLV as described in Section 6.6. The SR Segment List TLV is included for each of the SID-List(s) associated with the CP. Each SR Segment List TLV in turn includes SR Segment sub-TLV(s) to report the segment(s) and their status. The SR Segment List Metric sub-TLV is used to report the metric values and constraints for the specific SID List.

When the SR Policy CP is setup with the help of PCEP signaling then another MPLS-TE Policy State TLV MAY be used to encode the related PCEP objects corresponding to the LSP as defined in Section 5.2 specifically to report information and status that is not covered by the defined TLVs under Section 6. In the event of a conflict of information, the receiver MUST prefer the information originated via TLVs defined in Section 6 over the PCEP objects reported via the TE Policy State TLV.

8. Manageability Considerations

The Existing BGP operational and management procedures apply to this document. No new procedures are defined in this document. The considerations as specified in [RFC7752] apply to this document.
In general, it is assumed that the TE Policy head-end nodes are responsible for the distribution of TE Policy state information, while other nodes, e.g. the nodes in the path of a policy, MAY report the TE Policy information (if available) when needed. For example, the border routers in the inter-domain case will also distribute LSP state information since the ingress node may not have the complete information for the end-to-end path.

9. IANA Considerations

This document requires new IANA assigned codepoints.

9.1. BGP-LS NLRI-Types

IANA maintains a registry called "Border Gateway Protocol - Link State (BGP-LS) Parameters" with a sub-registry called "BGP-LS NLRI-Types".

The following codepoints have been assigned by early allocation process by IANA:

| Type | NLRI Type                  | Reference   |
|------|----------------------------+---------------|
| 5    | TE Policy NLRI type        | this document |

9.2. BGP-LS Protocol-IDs

IANA maintains a registry called "Border Gateway Protocol - Link State (BGP-LS) Parameters" with a sub-registry called "BGP-LS Protocol-IDs".

The following Protocol-ID codepoints have been assigned by early allocation process by IANA:

<table>
<thead>
<tr>
<th>Protocol-ID</th>
<th>NLRI information source protocol</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>RSVP-TE</td>
<td>this document</td>
</tr>
<tr>
<td>9</td>
<td>Segment Routing</td>
<td>this document</td>
</tr>
</tbody>
</table>

9.3. BGP-LS TLVs

IANA maintains a registry called "Border Gateway Protocol - Link State (BGP-LS) Parameters" with a sub-registry called "Node Anchor, Link Descriptor and Link Attribute TLVs".
The following TLV codepoints have been assigned by early allocation process by IANA:

<table>
<thead>
<tr>
<th>TLV Code</th>
<th>Description</th>
<th>Value defined in</th>
</tr>
</thead>
<tbody>
<tr>
<td>550</td>
<td>Tunnel ID TLV</td>
<td>this document</td>
</tr>
<tr>
<td>551</td>
<td>LSP ID TLV</td>
<td>this document</td>
</tr>
<tr>
<td>552</td>
<td>IPv4/6 Tunnel Head-end address TLV</td>
<td>this document</td>
</tr>
<tr>
<td>553</td>
<td>IPv4/6 Tunnel Tail-end address TLV</td>
<td>this document</td>
</tr>
<tr>
<td>554</td>
<td>SR Policy CP Descriptor TLV</td>
<td>this document</td>
</tr>
<tr>
<td>555</td>
<td>MPLS Local Cross Connect TLV</td>
<td>this document</td>
</tr>
<tr>
<td>556</td>
<td>MPLS Cross Connect Interface TLV</td>
<td>this document</td>
</tr>
<tr>
<td>557</td>
<td>MPLS Cross Connect FEC TLV</td>
<td>this document</td>
</tr>
<tr>
<td>1200</td>
<td>MPLS-TE Policy State TLV</td>
<td>this document</td>
</tr>
<tr>
<td>1201</td>
<td>SR BSID TLV</td>
<td>this document</td>
</tr>
<tr>
<td>1202</td>
<td>SR CP State TLV</td>
<td>this document</td>
</tr>
<tr>
<td>1203</td>
<td>SR CP Name TLV</td>
<td>this document</td>
</tr>
<tr>
<td>1204</td>
<td>SR CP Constraints TLV</td>
<td>this document</td>
</tr>
<tr>
<td>1205</td>
<td>SR Segment List TLV</td>
<td>this document</td>
</tr>
<tr>
<td>1206</td>
<td>SR Segment sub-TLV</td>
<td>this document</td>
</tr>
<tr>
<td>1207</td>
<td>SR Segment List Metric sub-TLV</td>
<td>this document</td>
</tr>
<tr>
<td>1208</td>
<td>SR Affinity Constraint sub-TLV</td>
<td>this document</td>
</tr>
<tr>
<td>1209</td>
<td>SR SRLG Constraint sub-TLV</td>
<td>this document</td>
</tr>
<tr>
<td>1210</td>
<td>SR Bandwidth Constraint sub-TLV</td>
<td>this document</td>
</tr>
<tr>
<td>1211</td>
<td>SR Disjoint Group Constraint sub-TLV</td>
<td>this document</td>
</tr>
<tr>
<td>TBD</td>
<td>SRv6 BSID TLV</td>
<td>this document</td>
</tr>
<tr>
<td>TBD</td>
<td>SR Policy Name TLV</td>
<td>this document</td>
</tr>
</tbody>
</table>

9.4. BGP-LS SR Policy Protocol Origin

This document requests IANA to maintain a new sub-registry under "Border Gateway Protocol - Link State (BGP-LS) Parameters". The new registry is called "SR Policy Protocol Origin" and contains the codepoints allocated to the "Protocol Origin" field defined in Section 4.5. The registry contains the following codepoints, with initial values, to be assigned by IANA:

<table>
<thead>
<tr>
<th>Code Point</th>
<th>Protocol Origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PCEP</td>
</tr>
<tr>
<td>2</td>
<td>BGP SR Policy</td>
</tr>
<tr>
<td>3</td>
<td>Local (via CLI, Yang model through NETCONF, gRPC, etc.)</td>
</tr>
</tbody>
</table>
9.5. BGP-LS TE State Object Origin

This document requests IANA to maintain a new sub-registry under "Border Gateway Protocol - Link State (BGP-LS) Parameters". The new registry is called "TE State Path Origin" and contains the codepoints allocated to the "Object Origin" field defined in Section 5. The registry contains the following codepoints, with initial values, to be assigned by IANA:

+----------+------------------+
<table>
<thead>
<tr>
<th>Code Point</th>
<th>Object Origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RSVP-TE</td>
</tr>
<tr>
<td>2</td>
<td>PCEP</td>
</tr>
<tr>
<td>3</td>
<td>Local/Static</td>
</tr>
</tbody>
</table>

9.6. BGP-LS TE State Address Family

This document requests IANA to maintain a new sub-registry under "Border Gateway Protocol - Link State (BGP-LS) Parameters". The new registry is called "TE State Address Family" and contains the codepoints allocated to the "Address Family" field defined in Section 5. The registry contains the following codepoints, with initial values, to be assigned by IANA:

+----------+-------------+
<table>
<thead>
<tr>
<th>Code Point</th>
<th>Address Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MPLS-IPv4</td>
</tr>
<tr>
<td>2</td>
<td>MPLS-IPv6</td>
</tr>
</tbody>
</table>

9.7. BGP-LS SR Segment Descriptors

This document requests IANA to maintain a new sub-registry under "Border Gateway Protocol - Link State (BGP-LS) Parameters". The new registry is called "SR Segment Descriptor Types" and contains the codepoints allocated to the "Segment Type" field defined in Section 6.8 and described in Section 6.8.1. The registry contains the following codepoints, with initial values, to be assigned by IANA:
## 9.8. BGP-LS Metric Type

This document requests IANA to maintain a new sub-registry under "Border Gateway Protocol - Link State (BGP-LS) Parameters". The new registry is called "Metric Type" and contains the codepoints allocated to the "metric type" field defined in Section 6.9. The registry contains the following codepoints, with initial values, to be assigned by IANA:

<table>
<thead>
<tr>
<th>Code Point</th>
<th>Metric Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>IGP Metric</td>
</tr>
<tr>
<td>1</td>
<td>Min Unidirectional Link Delay [RFC7471]</td>
</tr>
<tr>
<td>2</td>
<td>TE Metric [RFC3630]</td>
</tr>
</tbody>
</table>

## 10. Security Considerations

Procedures and protocol extensions defined in this document do not affect the BGP security model. See [RFC6952] for details.
11. Contributors

The following people have substantially contributed to the editing of this document:

Clarence Filsfils  
Cisco Systems  
Email: cfilsfil@cisco.com

12. Acknowledgements

The authors would like to thank Dhruv Dhody, Mohammed Abdul Aziz Khalid, Lou Berger, Acee Lindem, Siva Sivabalan, Arjun Sreekantiah, and Dhanendra Jain for their review and valuable comments.

13. References

13.1. Normative References

[I-D.ietf-idr-bgpls-srv6-ext]  

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

[RFC2119]  

[RFC2205]  

[RFC3209]  


13.2. Informative References


Authors’ Addresses

Stefano Previdi
Email: stefano@previdi.net

Ketan Talaulikar (editor)
Arrcus Inc
India
Email: ketant.ietf@gmail.com

Jie Dong (editor)
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China
Email: jie.dong@huawei.com

Mach(Guoyi) Chen
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China
Email: mach.chen@huawei.com
Abstract

Many Massively Scaled Data Centers (MSDCs) have converged on simplified layer 3 routing. Furthermore, requirements for operational simplicity have lead many of these MSDCs to converge on BGP as their single routing protocol for both their fabric routing and their Data Center Interconnect (DCI) routing. This document describes a solution which leverages BGP Link-State distribution and the Shortest Path First algorithm similar to Internal Gateway Protocols (IGPs) such as OSPF.

Status of This Memo

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

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

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

This Internet-Draft will expire on July 16, 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 (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

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 ........................................... 3
   1.1. BGP Shortest Path First (SPF) Motivation ........ 4
   1.2. Requirements Language .............................. 5
2. BGP Peering Models ..................................... 5
   2.1. BGP Single-Hop Peering on Network Node Connections .. 5
   2.2. BGP Peering Between Directly Connected Network Nodes .. 5
   2.3. BGP Peering in Route-Reflector or Controller Topology .. 6
3. BGP-LS Shortest Path Routing (SPF) SAFI .................. 6
4. Extensions to BGP-LS .................................... 6
   4.1. Node NLRI Usage and Modifications ................ 6
   4.2. Link NLRI Usage .................................. 7
   4.3. Prefix NLRI Usage ................................. 7
   4.4. BGP-LS Attribute Sequence-Number TLV ............... 8
5. Decision Process with SPF Algorithm .................... 9
   5.1. Phase-1 BGP NLRI Selection ....................... 9
   5.2. Dual Stack Support ................................ 10
   5.3. NEXT_HOP Manipulation ............................ 10
   5.4. NLRI Advertisement and Convergence ................. 10
   5.5. Error Handling .................................. 11
6. IANA Considerations .................................... 11
7. Security Considerations ............................... 12
   7.1. Acknowledgements ................................ 12
   7.2. Contributors .................................... 12
8. References ............................................ 12
1. Introduction

Many Massively Scaled Data Centers (MSDCs) have converged on simplified layer 3 routing. Furthermore, requirements for operational simplicity have lead many of these MSDCs to converge on BGP [RFC4271] as their single routing protocol for both their fabric routing and their Data Center Interconnect (DCI) routing. Requirements and procedures for using BGP are described in [RFC7938]. This document describes an alternative solution which leverages BGP-LS [RFC7752] and the Shortest Path First algorithm similar to Internal Gateway Protocols (IGPs) such as OSPF [RFC2328].

[RFC4271] defines the Decision Process that is used to select routes for subsequent advertisement by applying the policies in the local Policy Information Base (PIB) to the routes stored in its Adj-RIBs-In. The output of the Decision Process is the set of routes that are announced by a BGP speaker to its peers. These selected routes are stored by a BGP speaker in the speaker’s Adj-RIBs-Out according to policy.

[RFC7752] describes a mechanism by which link-state and TE information can be collected from networks and shared with external components using BGP. This is achieved by defining NLRI carried within BGP-LS AFI and BGP-LS SAFIs. The BGP-LS extensions defined in [RFC7752] makes use of the Decision Process defined in [RFC4271].

This document augments [RFC7752] by replacing its use of the existing Decision Process. The BGP-LS-SPF and BGP-LS-SPF-VPN API/Safi are introduced to insure backward compatibility. The Phase 1 and 2 decision functions of the Decision Process are replaced with the Shortest Path Algorithm (SPF) also known as the Dijkstra Algorithm. The Phase 3 decision function is also simplified since it is no longer dependent on the previous phases. This solution avails the benefits of both BGP and SPF-based IGPs. These include TCP based flow-control, no periodic link-state refresh, and completely incremental NLRI advertisement. These advantages can reduce the overhead in MSDCs where there is a high degree of Equal Cost Multi-Path (ECMPs) and the topology is very stable. Additionally, using a SPF-based computation can support fast convergence and the computation of Loop-Free Alternatives (LFAs) [RFC5286] in the event of link failures. Furthermore, a BGP based solution lends itself to multiple peering models including those incorporating route-reflectors [RFC4456] or controllers.
Support for Multiple Topology Routing (MTR) as described in [RFC4915] is an area for further study dependent on deployment requirements.

1.1. BGP Shortest Path First (SPF) Motivation

Given that [RFC7938] already describes how BGP could be used as the sole routing protocol in an MSDC, one might question the motivation for defining an alternate BGP deployment model when a mature solution exists. For both alternatives, BGP offers the operational benefits of a single routing protocol. However, BGP SPF offers some unique advantages above and beyond standard BGP distance-vector routing.

A primary advantage is that all BGP speakers in the BGP SPF routing domain will have a complete view of the topology. This will allow support of ECMP, IP fast-reroute (e.g., Loop-Free Alternatives), Shared Risk Link Groups (SRLGs), and other routing enhancements without advertisement of additional BGP paths or other extensions. In short, the advantages of an IGP such as OSPF [RFC2328] are availed in BGP.

With the simplified BGP decision process as defined in Section 5.1, NLRI changes can be disseminated throughout the BGP routing domain much more rapidly (equivalent to IGPs with the proper implementation).

Another primary advantage is a potential reduction in NLRI advertisement. With standard BGP distance-vector routing, a single link failure may impact 100s or 1000s prefixes and result in the withdrawal or re-advertisement of the attendant NLRI. With BGP SPF, only the BGP speakers corresponding to the link NLRI need withdraw the corresponding BGP-LS Link NLRI. This advantage will contribute to both faster convergence and better scaling.

With controller and route-reflector peering models, BGP SPF advertisement and distributed computation require a minimal number of sessions and copies of the NLRI since only the latest version of the NLRI from the originator is required. Given that verification of the adjacencies is done outside of BGP (see Section 2), each BGP speaker will only need as many sessions and copies of the NLRI as required for redundancy (e.g., one for SPF computation and another for backup). Functions such as Optimized Route Reflection (ORR) are supported without extension by virtue of the primary advantages. Additionally, a controller could inject topology that is learned outside the BGP routing domain.

Given that controllers are already consuming BGP-LS NLRI [RFC7752], reusing for the BGP-LS SPF leverages the existing controller implementations.
Another potential advantage of BGP SPF is that both IPv6 and IPv4 can be supported in the same address family using the same topology. Although not described in this version of the document, multi-topology extensions can be used to support separate IPv4, IPv6, unicast, and multicast topologies while sharing the same NLRI.

Finally, the BGP SPF topology can be used as an underlay for other BGP address families (using the existing model) and realize all the above advantages. A simplified peering model using IPv6 link-local addresses as next-hops can be deployed similar to [RFC5549].

1.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. BGP Peering Models

Depending on the requirements, scaling, and capabilities of the BGP speakers, various peering models are supported. The only requirement is that all BGP speakers in the BGP SPF routing domain receive link-state NLRI on a timely basis, run an SPF calculation, and update their data plane appropriately. The content of the Link NLRI is described in Section 4.2.

2.1. BGP Single-Hop Peering on Network Node Connections

The simplest peering model is the one described in section 5.2.1 of [RFC7938]. In this model, EBGP single-hop sessions are established over direct point-to-point links interconnecting the network nodes. For the purposes of BGP SPF, Link NLRI is only advertised if a single-hop BGP session has been established and the Link-State/SPF address family capability has been exchanged [RFC4790] on the corresponding session. If the session goes down, the NLRI will be withdrawn.

2.2. BGP Peering Between Directly Connected Network Nodes

In this model, BGP speakers peer with all directly connected network nodes but the sessions may be multi-hop and the direct connection discovery and liveliness detection for those connections are independent of the BGP protocol. How this is accomplished is outside the scope of this document. Consequently, there will be a single session even if there are multiple direct connections between BGP speakers. For the purposes of BGP SPF, Link NLRI is advertised as long as a BGP session has been established, the Link-State/SPF...
address family capability has been exchanged [RFC4790] and the corresponding link is up and considered operational.

2.3. BGP Peering in Route-Reflector or Controller Topology

In this model, BGP speakers peer solely with one or more Route Reflectors [RFC4456] or controllers. As in the previous model, direct connection discovery and liveliness detection for those connections are done outside the BGP protocol. For the purposes of BGP SPF, Link NLRI is advertised as long as the corresponding link is up and considered operational.

3. BGP-LS Shortest Path Routing (SPF) SAFI

In order to replace the Phase 1 and 2 decision functions of the existing Decision Process with an SPF-based Decision Process and streamline the Phase 3 decision functions in a backward compatible manner, this draft introduces a couple AFI/SAFIs for BGP LS SPF operation. The BGP-LS-SPF (AF 16388 / SAFI TBD1) and BGP-LS-SPF-VPN (AF 16388 / SAFI TBD2) [RFC4790] are allocated by IANA as specified in the Section 6.

4. Extensions to BGP-LS

[RFC7752] describes a mechanism by which link-state and TE information can be collected from networks and shared with external components using BGP protocol. It contains two parts: definition of a new BGP NLRI that describes links, nodes, and prefixes comprising IGP link-state information and definition of a new BGP path attribute (BGP-LS attribute) that carries link, node, and prefix properties and attributes, such as the link and prefix metric or auxiliary Router-IDs of nodes, etc.

The BGP protocol will be used in the Protocol-ID field specified in table 1 of [I-D.ietf-idr-bgpls-segment-routing-epe]. The local and remote node descriptors for all NLRI will be the BGP Router-ID (TLV 516) and either the AS Number (TLV 512) [RFC7752] or the BGP Confederation Member (TLV 517) [I-D.ietf-idr-bgpls-segment-routing-epe]. However, if the BGP Router-ID is known to be unique within the BGP Routing domain, it can be used as the sole descriptor.

4.1. Node NLRI Usage and Modifications

The SPF capability is a new Node Attribute TLV that will be added to those defined in table 7 of [RFC7752]. The new attribute TLV will only be applicable when BGP is specified in the Node NLRI Protocol ID.
field. The TBD TLV type will be defined by IANA. The new Node Attribute TLV will contain a single octet SPF algorithm field:

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--------------------------------------------------+
|              Type             |             Length            |
+--------------------------------------------------+
| SPF Algorithm |
+--------------------------------------------------+
```

The SPF Algorithm may take the following values:

1 - Normal SPF
2 - Strict SPF

When computing the SPF for a given BGP routing domain, only BGP nodes advertising the SPF capability attribute will be included the Shortest Path Tree (SPT).

4.2. Link NLRI Usage

The criteria for advertisement of Link NLRI are discussed in Section 2.

Link NLRI is advertised with local and remote node descriptors as described above and unique link identifiers dependent on the addressing. For IPv4 links, the links local IPv4 (TLV 259) and remote IPv4 (TLV 260) addresses will be used. For IPv6 links, the local IPv6 (TLV 261) and remote IPv6 (TLV 262) addresses will be used. For unnumbered links, the link local/remote identifiers (TLV 258) will be used. For links supporting having both IPv4 and IPv6 addresses, both sets of descriptors may be included in the same Link NLRI. The link identifiers are described in table 5 of [RFC7752].

The link IGP metric attribute TLV (TLV 1095) as well as any others required for non-SPF purposes SHOULD be advertised. Algorithms such as setting the metric inversely to the link speed as done in the OSPF MIB [RFC4750] may be supported. However, this is beyond the scope of this document.

4.3. Prefix NLRI Usage

Prefix NLRI is advertised with a local descriptor as described above and the prefix and length used as the descriptors (TLV 265) as described in [RFC7752]. The prefix metric attribute TLV (TLV 1155) as well as any others required for non-SPF purposes SHOULD be
advertised. For loopback prefixes, the metric should be 0. For non-loopback, the setting of the metric is beyond the scope of this document.

4.4. BGP-LS Attribute Sequence-Number TLV

A new BGP-LS Attribute TLV to BGP-LS NLRI types is defined to assure the most recent version of a given NLRI is used in the SPF computation. The TBD TLV type will be defined by IANA. The new BGP-LS Attribute TLV will contain an 8 octet sequence number. The usage of the Sequence Number TLV is described in Section 5.1.

Sequence Number

The 64-bit strictly increasing sequence number is incremented for every version of BGP-LS NLRI originated. BGP speakers implementing this specification MUST use available mechanisms to preserve the sequence number’s strictly increasing property for the deployed life of the BGP speaker (including cold restarts). One mechanism for accomplishing this would be to use the high-order 32 bits of the sequence number as a wrap/boot count that is incremented anytime the BGP Router router loses its sequence number state or the low-order 32 bits wrap.

When incrementing the sequence number for each self-originated NLRI, the sequence number should be treated as an unsigned 64-bit value. If the lower-order 32-bit value wraps, the higher-order 32-bit value should be incremented and saved in non-volatile storage. If by some chance the BGP Speaker is deployed long enough that there is a possibility that the 64-bit sequence number may wrap or a BGP Speaker completely loses its sequence number state (e.g., the BGP speaker hardware is replaced), the phase 1 decision function (see Section 5.1) rules should insure convergance, albeit, not immediately.
5. Decision Process with SPF Algorithm

The Decision Process described in [RFC4271] takes place in three distinct phases. The Phase 1 decision function of the Decision Process is responsible for calculating the degree of preference for each route received from a Speaker’s peer. The Phase 2 decision function is invoked on completion of the Phase 1 decision function and is responsible for choosing the best route out of all those available for each distinct destination, and for installing each chosen route into the Loc-RIB. The combination of the Phase 1 and 2 decision functions is also known as a Path vector algorithm.

When BGP-LS-SPF NLRI is received, all that is required is to determine whether it is the best-path by examining the Node-ID and sequence number as described in Section 5.1. If the best-path NLRI had changed, it will be advertised to other BGP-LS-SPF peers. If the attributes have changed (other than the sequence number), a BGP SPF calculation will be scheduled. However, a changed best-path can be advertised to other peer immediately and propagation of changes can approach IGP convergence times.

The SPF based Decision process starts with selecting only those Node NLRI whose SPF capability TLV matches with the local BGP speaker’s SPF capability TLV value. Since Link-State NLRI always contains the local descriptor [RFC7752], it will only be originated by a single BGP speaker in the BGP routing domain. These selected Node NLRI and their Link/Prefix NLRI are used to build a directed graph during the SPF computation. The best paths for BGP prefixes are installed as a result of the SPF process.

The Phase 3 decision function of the Decision Process [RFC4271] is also simplified since under normal SPF operation, a BGP speaker would advertise the NLRI selected for the SPF to all BGP peers with the BGP-LS/BGP-SPF AFI/SAFI. Application of policy would not be prevented but would normally not be necessary.

5.1. Phase-1 BGP NLRI Selection

The rules for NLRI selection are greatly simplified from [RFC4271].

1. If the NLRI is received from the BGP speaker originating the NLRI (as determined by the comparing BGP Router ID in the NLRI Node identifiers with the BGP speaker Router ID), then it is preferred over the same NLRI from non-originators.

2. If the Sequence-Number TLV is present in the BGP-LS Attribute, then the NLIR with the most recent, i.e., highest sequence number is selected. BGP-LS NLRI with a Sequence-Number TLV will be
considered more recent than NLRI without a BGP-LS or a BGP-LS Attribute that doesn’t include the Sequence-Number TLV.

3. The final tie-breaker is the NLRI from the BGP Speaker with the numerically largest BGP Router ID.

The modified Decision Process with SPF algorithm uses the metric from Link and Prefix NLRI Attribute TLVs [RFC7752]. As a result, any attributes that would influence the Decision process defined in [RFC4271] like ORIGIN, MULTI_EXIT_DISC, and LOCAL_PREF attributes are ignored by the SPF algorithm. Furthermore, the NEXT_HOP attribute value is preserved and validated but otherwise ignored during the SPF or best-path.

5.2. Dual Stack Support

The SPF based decision process operates on Node, Link, and Prefix NLRIs that support both IPv4 and IPv6 addresses. Whether to run a single SPF instance or multiple SPF instances for separate AFs is a matter of a local implementation. Normally, IPv4 next-hops are calculated for IPv4 prefixes and IPv6 next-hops are calculated for IPv6 prefixes. However, an interesting use-case is deployment of [RFC5549] where IPv6 link-local next-hops are calculated for both IPv4 and IPv6 prefixes. As stated in Section 1, support for Multiple Topology Routing (MTR) is an area for future study.

5.3. NEXT_HOP Manipulation

A BGP speaker that supports SPF extensions MAY interact with peers that don’t support SPF extensions. If the BGP Link-State address family is advertised to a peer not supporting the SPF extensions described herein, then the BGP speaker MUST conform to the NEXT_HOP rules mentioned in [RFC4271] when announcing the Link-State address family routes to those peers.

All BGP peers that support SPF extensions would locally compute the NEXT_HOP values as result of the SPF process. As a result, the NEXT_HOP attribute is always ignored on receipt. However BGP speakers should set the NEXT_HOP address according to the NEXT_HOP attribute rules mentioned in [RFC4271].

5.4. NLRI Advertisement and Convergence

A local failure will prevent a link from being used in the SPF calculation due to the IGP bi-directional connectivity requirement. Consequently, local link failures should always be given priority over updates (e.g., withdrawing all routes learned on a session) in order to ensure the highest priority propagation and optimal convergence.
Delaying the withdrawal of non-local routes is an area for further study as more IGP-like mechanisms would be required to prevent usage of stale NLRI.

5.5. Error Handling

When a BGP speaker receives a BGP Update containing a malformed SPF Capability TLV in the Node NLRI BGP-LS Attribute [RFC7752], it MUST ignore the received TLV and the Node NLRI and not pass it to other BGP peers as specified in [RFC7606]. When discarding a Node NLRI with malformed TLV, a BGP speaker SHOULD log an error for further analysis.

6. IANA Considerations

This document defines a couple AFI/SAFIs for BGP LS SPF operation and requests IANA to assign the BGP-LS-SPF AFI 16388 / SAFI TBD1 and the BGP-LS-SPF-VPN AFI 16388 / SAFI TBD2 as described in [RFC4750].

This document also defines two attribute TLV for BGP LS NLRI. We request IANA to assign TLVs for the SPF capability and the Sequence Number from the "BGP-LS Node Descriptor, Link Descriptor, Prefix Descriptor, and Attribute TLVs" Registry. Additionally, IANA is requested to create a new registry for "BGP-LS SPF Capability Algorithms" for the value of the algorithm both in the BGP-LS Node Attribute TLV and the BGP SPF Capability. The initial assignments are:

<table>
<thead>
<tr>
<th>Value(s)</th>
<th>Assignment Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved (not to be assigned)</td>
</tr>
<tr>
<td>1</td>
<td>SPF</td>
</tr>
<tr>
<td>2</td>
<td>Strict SPF</td>
</tr>
<tr>
<td>3-254</td>
<td>Unassigned (IETF Review)</td>
</tr>
<tr>
<td>255</td>
<td>Reserved (not to be assigned)</td>
</tr>
</tbody>
</table>

BGP-LS SPF Capability Algorithms
7. Security Considerations

This extension to BGP does not change the underlying security issues inherent in the existing [RFC4724] and [RFC4271].

7.1. Acknowledgements

The authors would like to thank .... for the review and comments.

7.2. Contributors

In addition to the authors listed on the front page, the following co-authors have contributed to the document.

Derek Yeung
Arrcus, Inc.
derek@arrcus.com

Abhay Roy
Cisco Systems
akr@cisco.com

Venu Venugopal
Cisco Systems
venuv@cisco.com

8. References

8.1. Normative References

[I-D.ietf-idr-bgpls-segment-routing-epe]


8.2. Information References


Authors’ Addresses

Keyur Patel
Arrcus, Inc.

Email: keyur@arrcus.com

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC  27513
USA

Email: acee@cisco.com

Shawn Zandi
Linkedin
222 2nd Street
San Francisco, CA  94105
USA

Email: szandi@linkedin.com

Gunter Van de Velde
Nokia
Antwerp
Belgium

Email: gunter.van_de_velde@nokia.com
BGP Flow Specification Filter Component for Time Constraints
draft-liang-idr-flowspec-v1-time-00.txt

Abstract

BGP flow specification version 1 (RFC5575) describes the distribution of traffic filter policy (traffic filters and actions) which are distributed via BGP to BGP peers to support the following 3 applications: (1) mitigation of Denial of Service (DoS), (2) traffic filtering in BGP/MPLS VPNs, and (3) centralized traffic control for networks with SDN or NFV controllers. A BGP Flow Filter that combines packet filter with time may provide an ability to for these three applications to have a flow filter operate for only a specific time.

This document proposes a new BGP Flow specification filter based on time.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 9, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

BGP flow specification [RFC5575] describes the distribution of filters and actions that apply when packets are received on a router with the flow specification function turned on. If one considers the reception of the packet as an event, then BGP [RFC4271] flow specification describes a set of minimalistic Event-MatchCondition-Action (ECA) policies were the match-condition is defined in the BGP NLRI, and the action is defined either by the default condition (accept traffic) or actions defined in Extended BGP Communities values [RFC4360].

The initial set of policy [RFC5575] for this policy includes 12 types of match filters encoded in two application specific AFI/SAFIs for the IPv4 API and the following SAFIs:

- IP traffic: AFI:1, SAFI, 133;

The 12 filters specified in [RFC5575] are "ANDED" and measured in a specific order. The packet does not match unless all filters match.

The popularity of these flow specification filters in deployment for the following applications has led to the requirement for more BGP
flow specification match filters in the NLRI and more BGP flow specification actions to support these applications

- mitigation of Denial of Service (DoS),
- support of traffic filtering in BGP/MPLS VPNs,
- centralized traffic control for networks with SDN or NFV controllers.

See [I-D.hares-idr-rfc5575bis] for additional details on these additional filters for BGP Flow Specification 1.

Since DDoS attacks are dynamic, redirection or filtering of a flow may be necessary only for some specified, and may be undesirable at other times. Thus network administrators may want to add a time filter to group of filters to be matched. For example, a network administrator may need to insert DoS filters for only a specific period while a DoS attack or a Distributed DoS (DDoS) attack is occurring. Another example, is the filter of traffic in the BGP/MPLS VPN to support prioritization of high priority services such as video traffic and limiting of bandwidth of low priority services (such as web browsing).

2. RFC 2119 language

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

3. Encoding of BGP-FS time

The encoding for BGP Flow Specification time

Type: Time Filter (TBD) Flow Specification Component type

Function: Match filter based on time.

Encoding: <type(1 octet), length(1 octet), <value>

value field: has the form shown in figure 3.
Starting Time: Expressed in seconds and microseconds since midnight (zero hour), January 1, 1970 (UTC). Precision of the "Starting Time" is implementation-dependent. If the "Starting Time Type" is set to 0, this field is invalid. An Invalid FlowSpecification filter is logged, and the NLRI ignored.

Duration: Expressed in seconds and microseconds. If this field is zero this filter is invalid. An Invalid FlowSpecification filter is logged, and the NLRI ignored.

4. IANA Considerations

This document requests IANA BGP allocations in line with [RFC7153].

This document requests IANA allocates an entry in the Flow Specification Component Types Registry with the following values:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Filter</td>
<td>TBD</td>
<td>This document.</td>
</tr>
</tbody>
</table>

5. Security Considerations

The time filter augments the other BGP Flow Filters with an indication of the time these filters are active. It is anticipated that these filters are deployed within secure BGP infrastructures and...
not in home environments. In home environments, the time of filters may provide insight to the activities of individuals. Anyone installing BGP Flow Filters in home environments should secure any flow filters by encrypting the data that flows over IP links.

6. References

6.1. Normative References


6.2. Informative References

Authors' Addresses

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI  48176
USA

Email: shares@ndzh.com

Qiandeng Liang
Huawei
101 Software Avenue, Yuhuatai District
Nanjing  210012
China

Email: liangqiandeng@huawei.com

Jianjie You
Huawei
101 Software Avenue, Yuhuatai District
Nanjing  210012
China

Email: youjianjie@huawei.com
BGP Flow Specification V2 Component for Time Constraints
draft-liang-idr-flowspec-v2-time-00.txt

Abstract

BGP flow specification version 1 (RFC5575) describes the distribution of traffic filter policy (traffic filters and actions) which are distributed via BGP to BGP peers to support the following 3 applications: (1) mitigation of Denial of Service (DoS), (2) traffic filtering in BGP/MPLS VPNs, and (3) centralized traffic control for networks with SDN or NFV controllers. A BGP Flow Filter that combines packet filter with time may provide an ability to for these three applications to have a flow filter operate for only a specific time. The traffic filtering and centralized traffic control applications may require user-defined ordering of filters rather than RFC5575’s defined order. BGP Flow Specification version 2016 allows for user ordering of flow specifications.

This document proposes a new BGP Flow specification filter for BGP Flow Specification 2.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 9, 2017.
1. Introduction

BGP flow specification [RFC5575] describes the distribution of filters and actions that apply when packets are received on a router with the flow specification function turned on. If one considers the reception of the packet as an event, then BGP [RFC4271] flow specification describes a set of minimalistic Event-MatchCondition-Action (ECA) policies were the match-condition is defined in the BGP NLRI, and the action is defined either by the default condition (accept traffic) or actions defined in Extended BGP Communities values [RFC4360].

The initial set of policy [RFC5575] for this policy includes 12 types of match filters encoded in two application specific AFI/SAFIs for the IPv4 AFI and the following SAFIs:

IP traffic: AFI:1, SAFI, 133;

BGP/MPLS VPN AFI:1 VPN SAFI, 134) for IPv4.
The 12 filters specified in [RFC5575] are "ANDED" and measured in a specific order. The packet does not match unless all filters match.

The popularity of these flow specification filters in deployment for the following applications has led to the requirement for more BGP flow specification match filters in the NLRI and more BGP flow specification actions to support these applications:

- mitigation of Denial of Service (DoS),
- support of traffic filtering in BGP/MPLS VPNs,
- centralized traffic control for networks with SDN or NFV controllers.

Since DDoS attacks are dynamic, redirection or filtering of a flow may be necessary only for some specified, and may be undesirable at other times. Thus network administrators may want to add a time filter to group of filters to be matched. For example, a network administrator may need to insert DoS filters for only a specific period while a DoS attack or a Distributed DoS (DDoS) attack is occurring. Another example, is the filter of traffic in the BGP/MPLS VPN to support prioritization of high priority services such as video traffic and limiting of bandwidth of low priority services (such as web browsing). A third example is centralized traffic control that varies traffic based on time of day.

Some of the requested BGP Flow Specification filters expand the number of filters and actions using the encoding rules described in [RFC5575] and [I-D.hares-idr-rfc5575bis]. Other requests for additional BGP Flow Specification filters request user-defined orders to BGP Flow Specification filters as described in [I-D.hares-idr-flowspec-v2].

This draft provides a timing filter for the user-ordered BGP Flow Specification filters (version 2).

2. RFC 2119 language

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

3. Encoding of BGP-FS time

The encoding for BGP Flow Specification time

Type: Time Filter (TBD) Flow Specification Component type
Function: Match filter based on time.

Encoding: <type(1 octet), length(1 octet), <value>>

value field: has the form shown in figure 3.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| NLRI Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Body per filter
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| order (2 octets) | type (2 octets) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Component length (2 octets) | time type | duration type |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Starting Time (seconds) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Starting Time (microseconds) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Duration (seconds) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Duration (microseconds) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Delay Time (seconds) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Delay Time (microseconds) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Periodic Time (seconds) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Periodic Time (microseconds) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Time filersub-TLV Format

Order: user define order of filter

Type: Time Flow Filter Component type (TBD)

Component length: Time Flow Filter Component length.

Time Type: Type of time filter with the values of:

* a) immediate start at start time (value 0),
b) delayed start (start time + Delay) (value 1), or

* c) period of time (from start time to duration time).

* Any other values cause this filter to be invalid.

Duration type: May be:

* a) normal (from start time until BGP flow specification is removed (value 0),

* b) time period (from start time until Duration time is completed),

* c) time period of Duration time of no traffic match after start time.

* Any other values cause this filter to be invalid.

Starting Time: Expressed in seconds and microseconds since midnight (zero hour), January 1, 1970 (UTC). Precision of the "Starting Time" is implementation-dependent. If the "Starting Time Type" is set to 0, this field is invalid.

Duration: Expressed in seconds and microseconds. If this field is zero this filter is invalid.

Delay: Expressed in seconds and microseconds. If this field is zero this filter is invalid.

An Invalid FlowSpecification filter is logged, and the NLRI ignored.

4. IANA Considerations

This document requests IANA BGP allocations in line with [RFC7153].

This document requests IANA allocates an entry in the Flow Specification Component Types Registry with the following values:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Filter v2</td>
<td>TBD</td>
<td>This document.</td>
</tr>
</tbody>
</table>

5. Security Considerations

The time filter augments the other BGP Flow Filters with an indication of the time these filters are active. It is anticipated that these filters are deployed within secure BGP infrastructures and
not in home environments. In home environments, the time of filters may provide insight to the activities of individuals. Anyone installing BGP Flow Filters in home environments should secure any flow filters by encrypting the data that flows over IP links.

6. References

6.1. Normative References

[I-D.hares-idr-flowspec-v2]

[I-D.hares-idr-rfc5575bis]


6.2. Informative References


Authors’ Addresses

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI  48176
USA

Email: shares@ndzh.com

Qiandeng Liang
Huawei
101 Software Avenue, Yuhuatai District
Nanjing  210012
China

Email: liangqiandeng@huawei.com

Jianjie You
Huawei
101 Software Avenue, Yuhuatai District
Nanjing  210012
China

Email: youjianjie@huawei.com
Advertising Segment Routing Policies in BGP
draft-previdi-idr-segment-routing-te-policy-07

Abstract

This document defines a new BGP SAFI with a new NLRI in order to advertise a candidate path of a Segment Routing Policy (SR Policy). An SR Policy is a set of candidate paths consisting of one or more segment lists. The headend of an SR Policy may learn multiple candidate paths for an SR Policy. Candidate paths may be learned via a number of different mechanisms, e.g., CLI, NetConf, PCEP, or BGP. This document specifies the way in which BGP may be used to distribute candidate paths. New sub-TLVs for the Tunnel Encapsulation Attribute are defined.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 24, 2017.
1. Introduction

Segment Routing (SR) allows a headend node to steer a packet flow along any path. Intermediate per-flow states are eliminated thanks to source routing [I-D.ietf-spring-segment-routing].

The headend node is said to steer a flow into an Segment Routing Policy (SR Policy).

The header of a packet steered in an SR Policy is augmented with the ordered list of segments associated with that SR Policy.

[I-D.filsfils-spring-segment-routing-policy] details the concepts of SR Policy and steering into an SR Policy. These apply equally to the MPLS and SRv6 instantiations of segment routing.

As highlighted in section 2 of [I-D.filsfils-spring-segment-routing-policy]:

- an SR policy may have multiple candidate paths learned via various mechanisms (CLI, NetConf, PCEP or BGP);
- the SRTE process selects the best candidate path for a Policy;
- the SRTE process binds a BSID to the selected path of the Policy;
- the SRTE process installs the selected path and its BSID in the forwarding plane.

This document specifies the way to use BGP to distribute one or more of the candidate paths of an SR policy to the headend of that policy. The SRTE process ([I-D.filsfils-spring-segment-routing-policy]) of the headend receives candidate paths from BGP, and possibly other sources as well, and the SRTE process then determines the selected path of the policy.

This document specifies a way of representing SR policies and their candidate paths in BGP UPDATE messages. BGP can then be used to propagate the SR policies and candidate paths. The usual BGP rules for BGP propagation and "bestpath selection" are used. At the headend of a specific policy, this will result in one or more candidate paths being installed into the "BGP table". These paths are then passed to the SRTE process. The SRTE process may compare
them to candidate paths learned via other mechanisms, and will choose one or more paths to be installed in the data plane. BGP itself does not install SRTE candidate paths into the data plane.

This document defines a new BGP address family (SAFI). In UPDATE messages of that address family, the NLRI identifies an SR policy, and the attributes specify candidate paths of that policy.

While for simplicity we may write that BGP advertises an SR Policy, it has to be understood that BGP advertises a candidate path of an SR policy and that this SR Policy might have several other candidate paths provided via BGP (via an NLRI with a different distinguisher as defined in this document), PCEP, NETCONF or local policy configuration.

Typically, a controller defines the set of policies and advertise them to policy head-end routers (typically ingress routers). The policy advertisement uses BGP extensions defined in this document. The policy advertisement is, in most but not all of the cases, tailored for a specific policy head-end. In this case the advertisement may sent on a BGP session to that head-end and not propagated any further.

Alternatively, a router (i.e.: an BGP egress router) advertises SR Policies representing paths to itself. In this case, it is possible to send the policy to each head-end over a BGP session to that head-end, without requiring any further propagation of the policy.

An SR Policy intended only for the receiver will, in most cases, not traverse any Route Reflector (RR, [RFC4456]).

In some situations, it is undesirable for a controller or BGP egress router to have a BGP session to each policy head-end. In these situations, BGP Route Reflectors may be used to propagate the advertisements, or it may be necessary for the advertisement to propagate through a sequence of one or more ASes. To make this possible, an attribute needs to be attached to the advertisement that enables a BGP speaker to determine whether it is intended to be a head-end for the advertised policy. This is done by attaching one or more Route Target Extended Communities to the advertisement ([RFC4360]).

The BGP extensions for the advertisement of SR Policies include following components:

- A new Subsequent Address Family Identifier (SAFI) whose NLRI identifies an SR Policy.
A set of new TLVs to be inserted into the Tunnel Encapsulation Attribute (as defined in [I-D.ietf-idr-tunnel-encaps]) specifying candidate paths of the SR policy, as well as other information about the SR policy.

One or more IPv4 address format route-target extended community ([RFC4360]) attached to the SR Policy advertisement and that indicates the intended head-end of such SR Policy advertisement.

The Color Extended Community (as defined in [I-D.ietf-idr-tunnel-encaps]) and used in order to steer traffic into an SR Policy, as described in [I-D.filsfils-spring-segment-routing-policy]. This document (Section 3) modifies the format of the Color Extended Community by using the two leftmost bits of the RESERVED field.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. SR TE Policy Encoding

2.1. SR TE Policy SAFI and NLRI

A new SAFI is defined: the SR Policy SAFI, (codepoint 73 assigned by IANA (see Section 8) from the "Subsequent Address Family Identifiers (SAFI) Parameters" registry).

The SR Policy SAFI uses a new NLRI defined as follows:

```
+------------------+
|  NLRI Length     | 1 octet
+------------------+
|  Distinguisher    | 4 octets
+------------------+
|  Policy Color     | 4 octets
+------------------+
|  Endpoint         | 4 or 16 octets
```

where:

- NLRI Length: 1 octet of length expressed in bits as defined in [RFC4760].
- Distinguisher: 4-octet value uniquely identifying the policy in the context of <color, endpoint> tuple. The distinguisher has no semantic value and is solely used by the SR Policy originator to make unique (from an NLRI perspective) multiple occurrences of the same SR Policy.

- Policy Color: 4-octet value identifying (with the endpoint) the policy. The color is used to match the color of the destination prefixes to steer traffic into the SR Policy [I-D.filsfils-spring-segment-routing-policy].

- Endpoint: identifies the endpoint of a policy. The Endpoint may represent a single node or a set of nodes (e.g., an anycast address or a summary address). The Endpoint is an IPv4 (4-octet) address or an IPv6 (16-octet) address according to the AFI of the NLRI.

The color and endpoint are used to automate the steering of BGP Payload prefixes on SR policy ([I-D.filsfils-spring-segment-routing-policy]).

The NLRI containing the SR Policy is carried in a BGP UPDATE message [RFC4271] using BGP multiprotocol extensions [RFC4760] with an AFI of 1 or 2 (IPv4 or IPv6) and with a SAFI of 73 (assigned by IANA from the "Subsequent Address Family Identifiers (SAFI) Parameters" registry).

An update message that carries the MP_REACH_NLRI or MP_UNREACH_NLRI attribute with the SR Policy SAFI MUST also carry the BGP mandatory attributes. In addition, the BGP update message MAY also contain any of the BGP optional attributes.

The next-hop of the SR Policy SAFI NLRI is set based on the AFI. For example, if the AFI is set to IPv4 (1), then the next-hop is encoded as a 4-byte IPv4 address. If the AFI is set to IPv6 (2), then the next-hop is encoded as a 16-byte IPv6 address of the router.

It is important to note that any BGP speaker receiving a BGP message with an SR Policy NLRI, will process it only if the NLRI is among the best paths as per the BGP best path selection algorithm. In other words, this document does not modify the BGP propagation or bestpath selection rules.

It has to be noted that if several candidate paths of the same SR Policy (endpoint, color) are signaled via BGP to a head-end, it is recommended that each NLRI use a different distinguisher. If BGP has installed into the BGP table two advertisements whose respective
NLRIs have the same color and endpoint, but different distinguishers, both advertisements are passed to the SRTE process.

2.2. SR TE Policy and Tunnel Encapsulation Attribute

The content of the SR Policy is encoded in the Tunnel Encapsulation Attribute originally defined in [I-D.ietf-idr-tunnel-encaps] using a new Tunnel-Type TLV (codepoint is 15, assigned by IANA (see Section 8) from the "BGP Tunnel Encapsulation Attribute Tunnel Types" registry).

The SR Policy Encoding structure is as follows:

SR Policy SAFI NLRI: <Distinguisher, Policy-Color, Endpoint>
Attributes:
   Tunnel Encaps Attribute (23)
      Tunnel Type: SR Policy
      Binding SID
      Preference
      Segment List
      Weight
      Segment
      Segment
      ...
      ...

where:

- SR Policy SAFI NLRI is defined in Section 2.1.
- Tunnel Encapsulation Attribute is defined in [I-D.ietf-idr-tunnel-encaps].
- Tunnel-Type is set to 15 (assigned by IANA from the "BGP Tunnel Encapsulation Attribute Tunnel Types" registry).
- Preference, Binding SID, Segment-List, Weight and Segment are defined in this document.
- Additional sub-TLVs may be defined in the future.

A Tunnel Encapsulation Attribute MUST NOT contain more than one TLV of type "SR Policy".

Multiple occurrences of "Segment List" MAY be encoded within the same SR Policy.

Multiple occurrences of "Segment" MAY be encoded within the same Segment List.
2.3. Remote Endpoint and Color

The Remote Endpoint and Color sub-TLVs, as defined in [I-D.ietf-idr-tunnel-encaps], MAY also be present in the SR Policy encodings.

If present, the Remote Endpoint sub-TLV MUST match the Endpoint of the SR Policy SAFI NLRI.

If present, the Color sub-TLV MUST match the Policy Color of the SR Policy SAFI NLRI.

2.4. SR TE Policy Sub-TLVs

This section defines the SR Policy sub-TLVs.

Preference, Binding SID, Segment-List are assigned from the "BGP Tunnel Encapsulation Attribute sub-TLVs" registry.

Weight and Segment Sub-TLVs are assigned from a new registry defined in this document and called: "SR Policy List Sub-TLVs". See Section 8 for the details of the registry.

2.4.1. Preference sub-TLV

The Preference sub-TLV does not have any effect on the BGP bestpath selection or propagation procedures. The contents of this sub-TLV are used by the SRTE process ([I-D.filsfils-spring-segment-routing-policy]).

The Preference sub-TLV is optional, MUST NOT appear more than once in the SR Policy and has following format:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Length      |     Flags     |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Preference (4 octets)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: TBD3 (to be assigned by IANA from the "BGP Tunnel Encapsulation Attribute sub-TLVs" registry).
Flags: 1 octet of flags. None are defined at this stage. Flags SHOULD be set to zero on transmission and MUST be ignored on receipt.

RESERVED: 1 octet of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.

Preference: a 4-octet value. The highest value is preferred.

2.4.2. SR TE Binding SID Sub-TLV

The Binding SID sub-TLV is not used by BGP. The contents of this sub-TLV are used by the SRTE process ([I-D.filsfils-spring-segment-routing-policy]).

The Binding SID sub-TLV is optional, MUST NOT appear more than once in the SR Policy and has the following format:

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Length      |     Flags     |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Binding SID (variable, optional)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: TBD4 (to be assigned by IANA from the "BGP Tunnel Encapsulation Attribute sub-TLVs" registry).

- Length: specifies the length of the value field not including Type and Length fields. Can be 2 or 6 or 18.

- Flags: 1 octet of flags. None are defined at this stage. Flags SHOULD be set to zero on transmission and MUST be ignored on receipt.

- RESERVED: 1 octet of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.

- Binding SID: if length is 2, then no Binding SID is present. If length is 6 then the Binding SID contains a 4-octet SID. If length is 10 then the Binding SID contains a 16-octet IPv6 SID.
2.4.3. Segment List Sub-TLV

The Segment List TLV encodes a single explicit path towards the endpoint. The Segment List sub-TLV includes the elements of the paths (i.e.: segments) as well as an optional Weight TLV.

The Segment List sub-TLV may exceed 255 bytes length due to large number of segments. Therefore a 2-octet length is required. According to [I-D.ietf-idr-tunnel-encaps], the first bit of the sub-TLV codepoint defines the size of the length field. Therefore, for the Segment List sub-TLV a code point of 128 (or higher) is used. See Section 8 for details of codepoints allocation.

The Segment List sub-TLV is mandatory and MAY appear multiple times in the SR Policy.

The Segment-List Sub-TLV MUST contain at least one Segment Sub-TLV and MAY contain a Weight Sub-TLV.

The Segment List sub-TLV has the following format:

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |             Length            |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                           sub-TLVs                          //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- **Type**: TBD5 (to be assigned by IANA from the "BGP Tunnel Encapsulation Attribute sub-TLVs" registry).
- **Length**: the total length (not including the Type and Length fields) of the sub-TLVs encoded within the Segment List sub-TLV.
- **RESERVED**: 1 octet of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.
- **sub-TLVs**:
  - An optional single Weight sub-TLV.
  - One or more Segment sub-TLVs.
2.4.3.1. Weight Sub-TLV

The Weight sub-TLV specifies the weight associated to a given candidate path (i.e.: a given segment list). The contents of this sub-TLV are used only by the SRTE process ([I-D.filsfils-spring-segment-routing-policy]).

The Weight sub-TLV is optional, MUST NOT appear more than once inside the Segment List sub-TLV, and has the following format:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Length      |     Flags     |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              Weight                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 9 (to be assigned by IANA from the registry "SR Policy List Sub-TLVs" defined in this document).

Length: 6.

Flags: 1 octet of flags. None are defined at this stage. Flags SHOULD be set to zero on transmission and MUST be ignored on receipt.

RESERVED: 1 octet of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.

2.4.3.2. Segment Sub-TLV

The Segment sub-TLV describes a single segment in a segment list (i.e., a single element of the explicit path). Multiple Segment sub-TLVs constitute an explicit path of the SR Policy.

The Segment sub-TLV is mandatory and MAY appear multiple times in the Segment List sub-TLV.

The Segment sub-TLV does not have any effect on the BGP bestpath selection or propagation procedures. The contents of this sub-TLV are used only by the SRTE process ([I-D.filsfils-spring-segment-routing-policy]).

[I-D.filsfils-spring-segment-routing-policy] defines several types of Segment Sub-TLVs:
Type 1: SID only, in the form of MPLS Label
Type 2: SID only, in the form of IPv6 address
Type 3: IPv4 Node Address with optional SID
Type 4: IPv6 Node Address with optional SID
Type 5: IPv4 Address + index with optional SID
Type 6: IPv4 Local and Remote addresses with optional SID
Type 7: IPv6 Address + index with optional SID
Type 8: IPv6 Local and Remote addresses with optional SID

2.4.3.2.1. Type 1: SID only, in the form of MPLS Label

The Type-1 Segment Sub-TLV encodes a single SID in the form of an
MPLS label. The format is as follows:

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Length      |     Flags     |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Label                        | TC  |S|       TTL     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- **Type**: 1 (to be assigned by IANA from the registry "SR Policy List
  Sub-TLVs" defined in this document).
- **Length**: 6.
- **Flags**: 1 octet of flags. None are defined at this stage. Flags
  SHOULD be set to zero on transmission and MUST be ignored on
  receipt.
- **Reserved**: 1 octet of reserved bits. SHOULD be unset on
  transmission and MUST be ignored on receipt.
- **Label**: 20 bits of label value.
- **TC**: 3 bits of traffic class.
- **S**: 1 bit of bottom-of-stack.
- **TTL**: 1 octet of TTL.

The following applies to the Type-1 Segment sub-TLV:

- The S bit SHOULD be zero upon transmission, and MUST be ignored
  upon reception.
o If the originator wants the receiver to choose the TC value, it sets the TC field to zero.

o If the originator wants the receiver to choose the TTL value, it sets the TTL field to 255.

o If the originator wants to recommend a value for these fields, it puts those values in the TC and/or TTL fields.

o The receiver MAY override the originator’s values for these fields. This would be determined by local policy at the receiver. One possible policy would be to override the fields only if the fields have the default values specified above.

2.4.3.2.2. Type 2: SID only, in the form of IPv6 address

The Type-2 Segment Sub-TLV encodes a single SID in the form of an IPv6 SID. The format is as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Length      |     Flags     |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             IPv6 SID (16 octets)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

o Type: 2 (to be assigned by IANA from the registry "SR Policy List Sub-TLVs" defined in this document).

o Length is 18.

o Flags: 1 octet of flags. None are defined at this stage. Flags SHOULD be set to zero on transmission and MUST be ignored on receipt.

o RESERVED: 1 octet of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.

o IPv6 SID: 16 octets of IPv6 address.

The IPv6 Segment Identifier (IPv6 SID) is defined in [I-D.ietf-6man-segment-routing-header].
2.4.3.2.3. Type 3: IPv4 Node Address with optional SID

The Type-3 Segment Sub-TLV encodes an IPv4 node address and an optional SID in the form of either an MPLS label or an IPv6 address. The format is as follows:

```
+-----------------+-----------------+-----------------+
|     Type     |   Length      |     Flags     |
+-----------------+-----------------+-----------------+
| IPv4 Node Address (4 octets) |
+-----------------+-----------------+-----------------+
   SID (optional, 4 or 16 octets)   //
```

where:

- **Type**: 3 (to be assigned by IANA from the registry "SR Policy List Sub-TLVs" defined in this document).
- **Length**: 6 or 10 or 22.
- **Flags**: 1 octet of flags. None are defined at this stage. Flags SHOULD be set to zero on transmission and MUST be ignored on receipt.
- **RESERVED**: 1 octet of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.
- **IPv4 Node Address**: a 4 octet IPv4 address representing a node.
- **SID**: either 4 octet MPLS SID or a 16 octet IPv6 SID.

The following applies to the Type-3 Segment sub-TLV:

- The IPv4 Node Address MUST be present.
- The SID is optional and MAY be of one of the following formats:
  - MPLS SID: a 4 octet label containing label, TC, S and TTL as defined in Section 2.4.3.2.1.
  - IPv6 SID: a 16 octet IPv6 address.
- If length is 6, then only the IPv4 Node Address is present.
o If length is 10, then the IPv4 Node Address and the MPLS SID are present.

o If length is 22, then the IPv4 Node Address and the IPv6 SID are present.

2.4.3.2.4. Type 4: IPv6 Node Address with optional SID

The Type-4 Segment Sub-TLV encodes an IPv6 node address and an optional SID in the form of either an MPLS label or an IPv6 address. The format is as follows:

```
+-----------------------------+-----------------------------+
| Type  | Length  | Flags | RESERVED |
+-----------------------------+-----------------------------+
// IPv6 Node Address (16 octets) //
// SID (optional, 4 or 16 octets) //
+-----------------------------+-----------------------------+
```

where:

o Type: 4 (to be assigned by IANA from the registry "SR Policy List Sub-TLVs" defined in this document).

o Length is 18 or 22 or 34.

o Flags: 1 octet of flags. None are defined at this stage. Flags SHOULD be set to zero on transmission and MUST be ignored on receipt.

o RESERVED: 1 octet of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.

o IPv6 Node Address: a 16 octet IPv6 address representing a node.

o SID: either 4 octet MPLS SID or a 16 octet IPv6 SID.

The following applies to the Type-4 Segment sub-TLV:

o The IPv6 Node Address MUST be present.

o The SID is optional and MAY be of one of the following formats:

* MPLS SID: a 4 octet label containing label, TC, S and TTL as defined in Section 2.4.3.2.1.
* IPV6 SID: a 16 octet IPv6 address.
  
  o If length is 18, then only the IPv6 Node Address is present.
  
  o If length is 22, then the IPv6 Node Address and the MPLS SID are present.
  
  o If length is 34, then the IPv6 Node Address and the IPv6 SID are present.

2.4.3.2.5. Type 5: IPv4 Address + Local Interface ID with optional SID

The Type-5 Segment Sub-TLV encodes an IPv4 node address, a local interface Identifier (Local Interface ID) and an optional SID in the form of either an MPLS label or an IPv4 address. The format is as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Length      |     Flags     |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Local Interface ID (4 octets)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 IPv4 Node Address (4 octets)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                SID (optional, 4 or 16 octets)               //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

where:

  o Type: 5 (to be assigned by IANA from the registry "SR Policy List Sub-TLVs" defined in this document).

  o Length is 10 or 14 or 26.

  o Flags: 1 octet of flags. None are defined at this stage. Flags SHOULD be set to zero on transmission and MUST be ignored on receipt.

  o RESERVED: 1 octet of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.

  o Local Interface ID: 4 octets as defined in [I-D.ietf-pce-segment-routing].

  o IPv4 Node Address: a 4 octet IPv4 address representing a node.
SID: either 4 octet MPLS SID or a 16 octet IPv6 SID.

The following applies to the Type-5 Segment sub-TLV:

- The IPv4 Node Address MUST be present.
- The Local Interface ID MUST be present.
- The SID is optional and MAY be of one of the following formats:
  * MPLS SID: a 4 octet label containing label, TC, S and TTL as defined in Section 2.4.3.2.1.
  * IPV6 SID: a 16 octet IPv6 SID.

- If length is 10, then the IPv4 Node Address and Local Interface ID are present.
- If length is 14, then the IPv4 Node Address, the Local Interface ID and the MPLS SID are present.
- If length is 26, then the IPv4 Node Address, the Local Interface ID and the IPv6 SID are present.

2.4.3.2.6. Type 6: IPv4 Local and Remote addresses with optional SID

The Type-6 Segment Sub-TLV encodes an adjacency local address, an adjacency remote address and an optional SID in the form of either an MPLS label or an IPv6 address. The format is as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Length      |     Flags     |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Local IPv4 Address (4 octets)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Remote IPv4 Address  (4 octets)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                     SID (4 or 16 octets)                    //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 6 (to be assigned by IANA from the registry "SR Policy List Sub-TLVs" defined in this document).
o Length is 10 or 14 or 26.

o Flags: 1 octet of flags. None are defined at this stage. Flags SHOULD be set to zero on transmission and MUST be ignored on receipt.

o RESERVED: 1 octet of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.

o Local IPv4 Address: a 4 octet IPv4 address.

o Remote IPv4 Address: a 4 octet IPv4 address.

o SID: either 4 octet MPLS SID or a 16 octet IPv6 SID.

The following applies to the Type-6 Segment sub-TLV:

- The Local IPv4 Address MUST be present and represents an adjacency local address.

- The Remote IPv4 Address MUST be present and represents the remote end of the adjacency.

- The SID is optional and MAY be of one of the following formats:
  * MPLS SID: a 4 octet label containing label, TC, S and TTL as defined in Section 2.4.3.2.1.
  * IPV6 SID: a 16 octet IPv6 address.

- If length is 10, then only the IPv4 Local and Remote addresses are present.

- If length is 14, then the IPv4 Local address, IPv4 Remote address and the MPLS SID are present.

- If length is 26, then the IPv4 Local address, IPv4 Remote address and the IPv6 SID are present.

2.4.3.2.7. Type 7: IPv6 Address + Local Interface ID with optional SID

The Type-7 Segment Sub-TLV encodes an IPv6 node address, a local interface identifier (Local Interface ID) and an optional SID in the form of either an MPLS label or an IPv6 address. The format is as follows:
where:

- **Type**: 7 (to be assigned by IANA from the registry "SR Policy List Sub-TLVs" defined in this document).
- **Length** is 22 or 26 or 38.
- **Flags**: 1 octet of flags. None are defined at this stage. Flags SHOULD be set to zero on transmission and MUST be ignored on receipt.
- **RESERVED**: 1 octet of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.
- **Local Interface ID**: 4 octets of interface index.
- **IPv6 Node Address**: a 16 octet IPv6 address representing a node.
- **SID**: either 4 octet MPLS SID or a 16 octet IPv6 SID.

The following applies to the Type-7 Segment sub-TLV:

- The IPv6 Node Address MUST be present.
- The Local Interface ID MUST be present.
- The SID is optional and MAY be of one of the following formats:
  - MPLS SID: a 4 octet label containing label, TC, S and TTL as defined in Section 2.4.3.2.1.
  - IPv6 SID: a 16 octet IPv6 address.
- If length is 22, then the IPv6 Node Address and Local Interface ID are present.
2.4.3.2.8. Type 8: IPv6 Local and Remote addresses with optional SID

The Type-8 Segment Sub-TLV encodes an adjacency local address, an adjacency remote address and an optional SID in the form of either an MPLS label or an IPv6 address. The format is as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Length      |     Flags     |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//               Local IPv6 Address (16 octets)                //
//               Remote IPv6 Address  (16 octets)              //
//                     SID (4 or 16 octets)                    //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- **Type**: 8 (to be assigned by IANA from the registry "SR Policy List Sub-TLVs" defined in this document).
- **Length**: 34 or 38 or 50.
- **Flags**: 1 octet of flags. None are defined at this stage. Flags SHOULD be set to zero on transmission and MUST be ignored on receipt.
- **RESERVED**: 1 octet of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.
- **Local IPv6 Address**: a 16 octet IPv6 address.
- **Remote IPv6 Address**: a 16 octet IPv6 address.
- **SID**: either 4 octet MPLS SID or a 16 octet IPv6 SID.

The following applies to the Type-8 Segment sub-TLV:
- The Local IPv6 Address MUST be present and represents an adjacency local address.

- The Remote IPv6 Address MUST be present and represents the remote end of the adjacency.

- The SID is optional and MAY be of one of the following formats:
  - MPLS SID: a 4 octet label containing label, TC, S and TTL as defined in Section 2.4.3.2.1.
  - IPV6 SID: a 16 octet IPv6 address.

- If length is 34, then only the IPv6 Local and Remote addresses are present.

- If length is 38, then the IPv6 Local address, IPv4 Remote address and the MPLS SID are present.

- If length is 50, then the IPv6 Local address, IPv4 Remote address and the IPv6 SID are present.

3. Extended Color Community

The Color Extended Community as defined in [I-D.ietf-idr-tunnel-encaps] is used to steer traffic into a policy.

When the Color Extended Community is used for the purpose of steering the traffic into an SRTE policy, the RESERVED field (as defined in [I-D.ietf-idr-tunnel-encaps]) is changed as follows:

```
1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|C 0|       RESERVED         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where CO bits are defined as the "Color-Only" bits.
[I-D.filsfils-spring-segment-routing-policy] defines the influence of these bits on the automated steering of BGP Payload traffic onto SRTE policies.

4. SR Policy Operations

As described in this document, the consumer of a SR Policy NLRI is not the BGP process. The BGP process is in charge of the origination and propagation of the SR Policy NLRI but its installation and use is
outside the scope of BGP ([I-D.filsfils-spring-segment-routing-policy]).

4.1. Configuration and Advertisement of SR TE Policies

Typically, but not limited to, an SR Policy is configured into a controller.

Multiple SR Policy NLRIs may be present with the same <color, endpoint> tuple but with different content when these SR policies are intended to different head-ends.

The distinguisher of each SR Policy NLRI prevents undesired BGP route selection among these SR Policy NLRIs and allow their propagation across route reflectors [RFC4456].

Moreover, one or more route-target SHOULD be attached to the advertisement, where each route-target identifies one or more intended head-ends for the advertised SR policy.

If no route-target is attached to the SR Policy NLRI, then it is assumed that the originator sends the SR Policy update directly (e.g., through a BGP session) to the intended receiver. In such case, the NO_ADVERTISE community MUST be attached to the SR Policy update.

4.2. Reception of an SR Policy NLRI

On reception of an SR Policy NLRI, a BGP speaker MUST determine if it’s first acceptable, then it determines if it is usable.

4.2.1. Acceptance of an SR Policy NLRI

When a BGP speaker receives an SR Policy NLRI from a neighbor it has to determine if it’s acceptable. The following applies:

- The SR Policy NLRI MUST include a distinguisher, color and endpoint field which implies that the length of the NLRI MUST be either 12 or 24 octets (depending on the address family of the endpoint). If the NLRI is not one of the legal lengths, a router supporting this document and that imports the route MUST consider it to be malformed and MUST apply the "treat-as-withdraw" strategy of [RFC7606].

- The SR Policy update MUST have either the NO_ADVERTISE community or at least one route-target extended community in IPv4-address format. If a router supporting this document receives an SR policy update with no route-target extended communities and no
NO_ADVERTISE community, the update MUST NOT be sent to the SRTE process. Furthermore, it SHOULD be considered to be malformed, and the "treat-as-withdraw" strategy of [RFC7606] applied.

- The Tunnel Encapsulation Attribute MUST be attached to the BGP Update and MUST have the Tunnel Type set to SR Policy (value to be assigned by IANA).
- Within the SR Policy NLRI, at least one Segment List sub-TLV MUST be present.
- Within the Segment List sub-TLV at least one Segment sub-TLV MUST be present.

A router that receives an SR Policy update that is not valid according to these criteria MUST treat the update as malformed. The route MUST NOT be passed to the SRTE process, and the "treat-as-withdraw" strategy of [RFC7606].

The Remote Endpoint and Color sub-TLVs, as defined in [I-D.ietf-idr-tunnel-encaps], MAY also be present in the SR Policy NLRI encodings. If present, the Remote Endpoint sub-TLV MUST match the Endpoint of the SR Policy SAFI NLRI. If they don’t match, the SR Policy advertisement MUST be considered as unacceptable. If present, the Color sub-TLV MUST match the Policy Color of the SR Policy SAFI NLRI. If they don’t match, the SR Policy advertisement MUST be considered as unacceptable.

A unacceptable SR Policy update that has a valid NLRI portion with invalid attribute portion MUST be considered as a withdraw of the SR Policy.

A unacceptable SR Policy update that has an invalid NLRI portion MUST trigger a reset of the BGP session.

4.2.2. Usable SR Policy NLRI

If one or more route-targets are present, then at least one route-target MUST match one of the BGP Identifiers of the receiver in order for the update to be considered usable. The BGP Identifier is defined in [RFC4271] as a 4 octet IPv4 address. Therefore the route-target extended community MUST be of the same format.

If one or more route-targets are present and no one matches any of the local BGP Identifiers, then, while the SR Policy NLRI is acceptable, it is not usable. It has to be noted that if the receiver has been explicitly configured to do so, it MAY propagate the SR Policy NLRI to its neighbors as defined in Section 4.2.4.
Usable SR Policy NLRIs are sent to the Segment Routing Traffic Engineering (SRTE) process. The description of the SRTE process is outside the scope of this document and it’s described in [I-D.filsfils-spring-segment-routing-policy].

4.2.3. Passing a usable SR Policy NLRI to the SRTE Process

Once BGP has determined that the SR Policy NLRI is usable, BGP passes the path to the SRTE process ([I-D.filsfils-spring-segment-routing-policy]).

The SRTE process applies the rules defined in [I-D.filsfils-spring-segment-routing-policy] to determine whether a path is valid and to select the best path among the valid paths.

4.2.4. Propagation of an SR Policy

By default, a BGP node receiving an SR Policy NLRI MUST NOT propagate it to any EBGP neighbor.

However, a node MAY be explicitly configured to advertise a received SR Policy NLRI to neighbors according to normal BGP rules (i.e., EBGP propagation by an ASBR or iBGP propagation by a Route-Reflector).

SR Policy NLRIs that have been determined acceptable and valid can be propagated, even the ones that are not usable.

Only SR Policy NLRIs that do not have the NO_ADVERTISE community attached to them can be propagated.

4.3. Flowspec and SR Policies

The SR Policy can be carried in context of a Flowspec NLRI ([RFC5575]). In this case, when the redirect to IP next-hop is specified as in [I-D.ietf-idr-flowspec-redirect-ip], the tunnel to the next-hop is specified by the segment list in the Segment List sub-TLVs. The Segment List (e.g., label stack or IPv6 segment list) is imposed to flows matching the criteria in the Flowspec route to steer them towards the next-hop as specified in the SR Policy SAFI NLRI.

5. Contributors

Arjun Sreekantiah
Cisco Systems
US

Email: asreekan@cisco.com
6. Acknowledgments

The authors of this document would like to thank Shyam Sethuram and John Scudder for their comments and review of this document.

7. Implementation Status

Note to RFC Editor: Please remove this section prior to publication, as well as the reference to RFC 7942.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.
According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

Several early implementations exist and will be reported in detail in a forthcoming version of this document. For purposes of early interoperability testing, when no FCFS code point was available, implementations have made use of the following values:

- Preference sub-TLV: 6
- Binding SID sub-TLV: 7
- Segment List sub-TLV: 128

When IANA-assigned values are available, implementations will be updated to use them.

8. IANA Considerations

This document defines new Sub-TLVs in following existing registries:

- Subsequent Address Family Identifiers (SAFI) Parameters
- BGP Tunnel Encapsulation Attribute Tunnel Types
- BGP Tunnel Encapsulation Attribute sub-TLVs

This document also defines a new registry: "SR Policy List Sub-TLVs".

8.1. Existing Registry: Subsequent Address Family Identifiers (SAFI) Parameters

This document defines a new SAFI in the registry "Subsequent Address Family Identifiers (SAFI) Parameters" that has been assigned by IANA:

<table>
<thead>
<tr>
<th>Codepoint</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>73</td>
<td>SR Policy SAFI</td>
<td>This document</td>
</tr>
</tbody>
</table>

8.2. Existing Registry: BGP Tunnel Encapsulation Attribute Tunnel Types

This document defines a new Tunnel-Type in the registry "BGP Tunnel Encapsulation Attribute Tunnel Types" that has been assigned by IANA:
8.3. Existing Registry: BGP Tunnel Encapsulation Attribute sub-TLVs

This document defines new sub-TLVs in the registry "BGP Tunnel Encapsulation Attribute sub-TLVs" to be assigned by IANA:

<table>
<thead>
<tr>
<th>Codepoint</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD3</td>
<td>Preference sub-TLV</td>
<td>This document</td>
</tr>
<tr>
<td>TBD4</td>
<td>Binding SID sub-TLV</td>
<td>This document</td>
</tr>
<tr>
<td>TBD5</td>
<td>Segment List sub-TLV</td>
<td>This document</td>
</tr>
</tbody>
</table>

8.4. New Registry: SR Policy List Sub-TLVs

This document defines a new registry called "SR Policy List Sub-TLVs". The allocation policy of this registry is "First Come First Served (FCFS)" according to [RFC5226].

Following Sub-TLV codepoints are defined:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MPLS SID sub-TLV</td>
<td>This document</td>
</tr>
<tr>
<td>2</td>
<td>IPv6 SID sub-TLV</td>
<td>This document</td>
</tr>
<tr>
<td>3</td>
<td>IPv4 Node and SID sub-TLV</td>
<td>This document</td>
</tr>
<tr>
<td>4</td>
<td>IPv6 Node and SID sub-TLV</td>
<td>This document</td>
</tr>
<tr>
<td>5</td>
<td>IPv4 Node, index and SID sub-TLV</td>
<td>This document</td>
</tr>
<tr>
<td>6</td>
<td>IPv4 Local/Remote addresses and SID sub-TLV</td>
<td>This document</td>
</tr>
<tr>
<td>7</td>
<td>IPv6 Node, index and SID sub-TLV</td>
<td>This document</td>
</tr>
<tr>
<td>8</td>
<td>IPv6 Local/Remote addresses and SID sub-TLV</td>
<td>This document</td>
</tr>
<tr>
<td>9</td>
<td>Weight sub-TLV</td>
<td>This document</td>
</tr>
</tbody>
</table>

9. Security Considerations
TBD.

10. References

10.1. Normative References

[I-D.ietf-idr-tunnel-encaps]
10.2. Informational References

[I-D.filsfils-spring-segment-routing-policy]

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

[I-D.ietf-idr-flowspec-redirect-ip]

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


Authors' Addresses
Stefano Previdi (editor)
Cisco Systems, Inc.
IT
Email: stefano@previdi.net