Inter-Domain Routing S. Previdi
Internet-Draft
Intended status: Standards Track K. Talaulikar, Ed.
Expires: 29 July 2023 Cisco Systems
J. Dong, Ed.
M. Chen
Huawei Technologies
H. Gredler
RtBrick Inc.
J. Tantsura
Microsoft
25 January 2023

Distribution of Traffic Engineering (TE) Policies and State using BGP-LS
draft-ietf-idr-te-lsp-distribution-19

Abstract

This document describes a mechanism to collect the Traffic Engineering 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 29 July 2023.

Copyright Notice

Copyright (c) 2023 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

In many network environments, traffic engineering (TE) policies are instantiated into various forms:

* MPLS Traffic Engineering Label Switched Paths (TE-LSPs).

* Segment Routing (SR) Policies as defined in [RFC9256]

* 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 policies: MPLS TE LSPs, SR Policies, etc.

SR Policy architecture details are specified in [RFC9256]. An SR Policy comprises 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 document covers the advertisement of state information at the individual SR Policy CP level.

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 optimization. While some extensions are proposed in the 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 a TE Policy state collection mechanism complementary to the mechanism defined in [RFC8231].

```
                                 -----------
                                |   -----   |
            Service             |  | TED |<-+------------>
            Request             |   -----   |  TED synchronization
               |                |     |     |  mechanism (e.g.,
               v                |     |     |  routing protocol)
         ------------- Request/ |     v     |
        |             | Response|   -----   |
        |     NMS     |<--------+> | PCE |  |
        |             |         |   -----   |
         -------------           -----------
       Service |
       Request |
               v
          ----------  Signaling   ----------
         | Head-End | Protocol   | Adjacent |
         |  Node    |<---------->|   Node   |
          ----------              ----------
```

Figure 1. Management-Based PCE Usage

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 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 of both meeting the application's requirements and utilizing 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 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 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |            NLRI Type          |     Total NLRI Length         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    //                  Link-State NLRI (variable)                 //
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Additional "NLRI Types" are defined for TE Policy Information as following:

* SR Policy Candidate Path NLRI (value 5)

* MPLS-TE LSP NLRI (value TBD)

* MPLS Local Cross-connect NLRI (value TBD)

The SR Policy Candidate Path NLRI is used to report the state details of individual SR Policy Candidate paths along with their underlying segment lists.

The common format for these NLRI types is defined in Section 3 below.

3. TE Policy NLRI Types

This document defines TE Policy NLRI Types with their common 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 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+
    |  Protocol-ID  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                           Identifier                          |
    |                           (64 bits)                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //            Node Descriptor TLV (for the Headend)            //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //                TE Policy Descriptors (variable)             //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

* Protocol-ID field specifies the component that owns the TE Policy state in the advertising node. The existing protocol-id value of 5 for Static Configuration applies for some of the NLRI types and the following additional Protocol-IDs are defined for some of the other types in this document:

Protocol-ID	NLRI information source protocol
8	RSVP-TE
9	Segment Routing

* "Identifier" is an 8 octet value as defined in [RFC7752].

* "Local Node Descriptor" (TLV 256) as defined in [RFC7752] that describes the headend node.

* "TE Policy Descriptors" consists of one or more of the TLVs listed as below for use with the respective NLRI type advertisements as specified in Section 4:

Codepoint	Descriptor TLVs
550	Tunnel ID
551	LSP ID
552	IPv4/6 Tunnel Head-end address
553	IPv4/6 Tunnel Tail-end address
554	SR Policy Candidate Path
555	Local MPLS Cross Connect

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 at least one of 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:

* BGP Confederation Member (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 NLRI types defined in Section 3.

4.1. Tunnel Identifier

The Tunnel Identifier TLV contains the Tunnel ID defined in [RFC3209] and is used with the Protocol-ID set to RSVP-TE to advertise the MPLS-TE LSP NLRI Type. It is a mandatory TE policy descriptor TLV for MPLS-TE LSP NLRI type. It 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tunnel ID             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

* Type: 550

* Length: 2 octets.

* Tunnel ID: 2 octets as defined in [RFC3209].

4.2. LSP Identifier

The LSP Identifier TLV contains the LSP ID defined in [RFC3209] and is used with the Protocol-ID set to RSVP-TE to advertise the MPLS-TE LSP NLRI Type. It is a mandatory TE policy descriptor TLV for MPLS-TE LSP NLRI type. It 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              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            LSP ID             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 with the Protocol-ID set to RSVP-TE to advertise the MPLS-TE LSP NLRI Type. It is a mandatory TE policy descriptor TLV for MPLS-TE LSP NLRI type. It 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              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //        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 with the Protocol-ID set to RSVP-TE to advertise the MPLS-TE LSP NLRI Type. It is a mandatory TE policy descriptor TLV for MPLS-TE LSP NLRI type. It 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             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//        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 [RFC9256]. It is used with the Protocol-ID set to Segment Routing to advertise the SR Policy Candidate Path NLRI Type. It is a mandatory TE policy descriptor TLV for SR Policy Candidate Path NLRI type. 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 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |             Type              |             Length            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |Protocol-origin|     Flags     |          RESERVED             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                    Endpoint (4 or 16 octets)                 //
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                    Policy Color (4 octets)                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |              Originator AS Number (4 octets)                  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |             Originator Address (4 or 16 octets)              //
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                    Discriminator (4 octets)                   |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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.

Code Point	Protocol Origin
1	PCEP
2	BGP SR Policy
3	Configuration (CLI, YANG model via NETCONF, etc.)

* Flags: 1-octet field with following bit positions defined. Other bits MUST be cleared by the originator and MUST be ignored by a receiver.

```
        0 1 2 3 4 5 6 7
       +-+-+-+-+-+-+-+-+
       |E|O|           |
       +-+-+-+-+-+-+-+-+
```

where:

- E-Flag: Indicates the encoding of endpoint as IPv6 address when set and IPv4 address when clear

- O-Flag: Indicates the encoding of originator address as IPv6 address when set and IPv4 address when clear

* Reserved: 2 octets which MUST be set to 0 by the originator and MUST be ignored by a receiver.

* Endpoint: 4 or 16 octets (as indicated by the flags) containing the address of the endpoint of the SR Policy

* Color: 4 octets that indicate the color of the SR Policy

* Originator ASN: 4 octets to carry the 4-byte encoding of the ASN of the originator. Refer to section 2.4 of [RFC9256] for details.

* Originator Address: 4 or 16 octets (as indicated by the flags) to carry the address of the originator. Refer to section 2.4 of [RFC9256] for details.

* Discriminator: 4 octets to carry the discriminator of the path. Refer to section 2.5 of [RFC9256] for details.

4.6. Local MPLS Cross Connect

The Local MPLS Cross Connect TLV identifies a local MPLS state in the form of an incoming label and interface followed by an outgoing label and interface. The 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]. It is a mandatory TE policy descriptor TLV for MPLS Local Cross-connect NLRI type.

The Local MPLS Cross Connect 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Incoming label (4 octets)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Outgoing label (4 octets)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                         Sub-TLVs (variable)                 //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 an FEC Sub-TLV.

The following sub-TLVs are defined for the Local MPLS Cross Connect TLV:

Codepoint	Descriptor TLV
556	MPLS Cross Connect Interface
557	MPLS Cross Connect FEC

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
 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     |
+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          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 with 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 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Type              |            Length             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Flags       |  Masklength   |   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.

* Prefix: an IPv4 or IPv6 prefix whose mask length is given by the " Mask Length" field padded to an octet boundary.

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 LSP NLRI type and it is carried in the optional non-transitive BGP Attribute "LINK_STATE Attribute" defined in [RFC7752]. These MPLS-TE LSP characteristics include the characteristics and attributes of the LSP, 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 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Type            |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Object-origin | Address Family|            RESERVED           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//        MPLS-TE Policy State Objects (variable)              //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Figure 1: MPLS-TE Policy State TLV

* Type: 1200

* Length: the total length of the MPLS-TE Policy State TLV not including the 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. The following object-origin codepoints are defined in this document.

Code Point	Object Origin
1	RSVP-TE
2	PCEP
3	Local/Static

* Address Family: describes the address family used to set up the MPLS-TE policy. The following address family values are defined in this document:

Code Point	Dataplane
1	MPLS-IPv4
2	MPLS-IPv6

* RESERVED: 16-bit field. SHOULD be set to 0 on transmission and MUST be ignored on receipt.

* 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]

* SESSION_ATTRIBUTE [RFC3209]

* EXPLICIT_ROUTE Object (ERO) [RFC3209]

* ROUTE_RECORD Object (RRO) [RFC3209]

* FAST_REROUTE Object [RFC4090]

* DETOUR Object [RFC4090]

* EXCLUDE_ROUTE Object (XRO) [RFC4874]

* SECONDARY_EXPLICIT_ROUTE Object (SERO) [RFC4873]

* SECONDARY_RECORD_ROUTE (SRRO) [RFC4873]

* LSP_ATTRIBUTES Object [RFC5420]

* LSP_REQUIRED_ATTRIBUTES Object [RFC5420]

* PROTECTION Object [RFC3473][RFC4872][RFC4873]

* ASSOCIATION Object [RFC4872]

* PRIMARY_PATH_ROUTE Object [RFC4872]

* ADMIN_STATUS Object [RFC3473]

* LABEL_REQUEST Object [RFC3209][RFC3473]

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 consist 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

This section defines the various TLVs which enable the headend to report the state at the SR Policy CP level. These TLVs (and their sub-TLVs) are carried in the optional non-transitive BGP Attribute "LINK_STATE Attribute" defined in [RFC7752] associated with the SR Policy CP NLRI type.

The detailed procedures for the advertisement are described in Section 7.

6.1. SR Binding SID TLV

The SR Binding SID (BSID) is an optional TLV that is used to report 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 [RFC9256]. Only a single instance of this TLV is advertised for a given CP. If multiple instances are present, then the first one is considered valid and the rest are ignored.

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 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 [RFC9256]. Other bits MUST be cleared by the originator and MUST be ignored by a 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. MUST be set to 0 by the originator and MUST be ignored by a receiver.

* Binding SID: It indicates the operational or allocated BSID value 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.

The BSID fields above are 4-octet carrying the MPLS Label or 16-octet 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 MUST be set to 0 by the originator and MUST be ignored by a receiver.

```
   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
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |          Label                        | TC  |S|       TTL     |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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, which enables the specification of the SRv6 Endpoint Behavior, be used for signaling of an SRv6 Binding SID.

6.2. SRv6 Binding SID TLV

The SRv6 Binding SID (BSID) is an optional TLV that is used to report 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 [RFC9256]. Multiple instances of this TLV may be used to report each of the SRv6 BSIDs associated with the CP.

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 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:

* Type: 1212

* 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 [RFC9256]. Other bits MUST be cleared by the originator and MUST be ignored by a receiver.

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|B|U|F|                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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. MUST be set to 0 by the originator and MUST be ignored by a receiver.

* Binding SID: It indicates the operational or allocated BSID value 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.

* 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 TLV

The SR Candidate Path (CP) State TLV provides the operational status and attributes of the SR Policy at the CP level. Only a single instance of this TLV is advertised for a given CP. If multiple instances are present, then the first one is considered valid and the rest are ignored.

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 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 [RFC9256].

* RESERVED: 1 octet. MUST be set to 0 by the originator and MUST be ignored by a 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 [RFC9256]. Other bits MUST be cleared by the originator and MUST be ignored by a receiver.

```
 0                   1
 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|U|         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- S-Flag: Indicates the CP is in an 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 a local template when set. Refer to section 8.5 of [RFC9256] for details.

- 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 the best preference amongst the available CPs. Refer to section 8.2 of [RFC9256] for details.

   - T-Flag: Indicates the CP has been marked as eligible for use as Transit Policy on the headend when set. Refer to section 8.3 of [RFC9256].

   - U-Flag: Indicates the SR Policy that the CP belongs to is dropping traffic as a result of the "drop upon invalid" behavior being activated.

* Preference: 4-octet value which indicates the preference of the CP. Refer to section 2.7 of [RFC9256] for details.

6.4. SR Policy Name TLV

The SR Policy Name TLV is an optional TLV that is used to carry the symbolic name associated with the SR Policy. Only a single instance of this TLV is advertised for a given CP. If multiple instances are present, then the first one is considered valid and the rest are ignored.

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

where:

* Type: 1213

* Length: variable

* SR Policy Name: Symbolic name for the SR Policy without a NULL terminator as specified in section 2.6 of [RFC9256]. 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 TLV

The SR Candidate Path Name TLV is an optional TLV that is used to carry the symbolic name associated with the candidate path. Only a single instance of this TLV is advertised for a given CP. If multiple instances are present, then the first one is considered valid and the rest are ignored.

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

where:

* Type: 1203

* Length: variable

* Candidate Path Name: Symbolic name for the SR Policy candidate path without a NULL terminator as specified in section 2.6 of [RFC9256]. 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 TLV

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 either by the headend or may be delegated to a controller. The constraints may also be applied to an explicit path where the computation entity is expected to validate that the path satisfies the specified constraints and if not the path is to be invalidated (e.g., due to topology changes). Only a single instance of this TLV is advertised for a given CP. If multiple instances are present, then the first one is considered valid and the rest are ignored.

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 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Type            |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Flags            |          RESERVED1            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             MTID              |  Algorithm    |  RESERVED2    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   sub-TLVs (variable)                                        //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

* Type: 1204

* Length: variable

* 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 MUST be cleared by the originator and MUST be ignored by a receiver.

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|D|P|U|A|T|S|F|                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- D-Flag: Indicates that the CP uses SRv6 dataplane when set and SR/MPLS dataplane when clear

- P-Flag: Indicates that the CP prefers the use of only protected SIDs when set. This flag is mutually exclusive with the U-Flag.

- U-Flag: Indicates that the CP prefers the use of only unprotected SIDs when set. This flag is mutually exclusive with the P-Flag.

- A-Flag: Indicates that the CP uses only the SIDs belonging to the specified SR Algorithm when set

- T-Flag: Indicates that the CP uses only the SIDs belonging to the specified topology when set

- S-Flag: Indicates that the use of protected (P-Flag) or unprotected (U-Flag) SIDs becomes a strict constraint instead of a preference when set

- F-Flag: Indicates that the CP is fixed once computed and not modified except on operator intervention.

* RESERVED1: 2 octets. MUST be set to 0 by the originator and MUST be ignored by a receiver.

* MTID: Indicates the multi-topology identifier of the IGP topology that is preferred to be used when the path is set up. When the T-flag is set then the path is strictly using the specified topology SIDs only.

* Algorithm: Indicates the algorithm that is preferred to be used when the path is set up. When the A-flag is set then the path is strictly using the specified algorithm SIDs only.

* RESERVED2: 1 octet. MUST be set to 0 by the originator and MUST be ignored by a 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 Sub-TLV

The SR Affinity Constraint sub-TLV is an optional sub-TLV of the SR CP Constraints 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]. Only a single instance of this sub-TLV is advertised for a given CP. If multiple instances are present, then the first one is considered valid and the rest are ignored.

The 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 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. MUST be set to 0 by the originator and MUST be ignored by a receiver.

* Exclude-Any EAG: the bitmask used to represent the affinities that have been excluded from the path.

* Include-Any EAG: the bitmask used to represent the affinities that have been included in the path.

* Include-All EAG: the bitmask used to represent all the affinities that have been included in the path.

6.6.2. SR SRLG Constraint Sub-TLV

The SR SRLG Constraint sub-TLV is an optional sub-TLV of the SR CP Constraints TLV that is used to carry the Shared Risk Link Group (SRLG) values [RFC4202] that have been excluded from the candidate path. Only a single instance of this sub-TLV is advertised for a given CP. If multiple instances are present, then the first one is considered valid and the rest are ignored.

The 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 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               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 Sub-TLV

The SR Bandwidth Constraint sub-TLV is an optional sub-TLV of the SR CP Constraints TLV that is used to indicate the bandwidth that has been requested for the candidate path. Only a single instance of this sub-TLV is advertised for a given CP. If multiple instances are present, then the first one is considered valid and the rest are ignored.

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

where:

* Type: 1210

* Length: 4 octets

* Bandwidth: 4 octets which specify the desired bandwidth in unit of bytes per second in IEEE floating point format.

6.6.4. SR Disjoint Group Constraint Sub-TLV

The SR Disjoint Group Constraint sub-TLV is an optional sub-TLV of the SR CP Constraints 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. Only a single instance of this sub-TLV is advertised for a given CP. If multiple instances are present, then the first one is considered valid and the rest are ignored.

The 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 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Type            |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Request-Flags | Status-Flags  |            RESERVED           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Disjoint Group Identifier                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 MUST be cleared by the originator and MUST be ignored by a receiver.

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|S|N|L|F|I|     |
+-+-+-+-+-+-+-+-+
```

where:

- S-Flag: Indicates that SRLG disjointness is requested when set

- N-Flag: Indicates that node disjointness is requested when set

- L-Flag: Indicates that link disjointness is requested when set

- F-Flag: Indicates that the computation may fallback to a lower level of disjointness amongst the ones requested when all cannot be achieved when set

- 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 when set

* 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 MUST be cleared by the originator and MUST be ignored by a receiver.

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|S|N|L|F|I|X|   |
+-+-+-+-+-+-+-+-+
```

where:

- S-Flag: Indicates that SRLG disjointness is achieved when set

- N-Flag: Indicates that node disjointness is achieved when set

- L-Flag: Indicates that link disjointness is achieved when set

- F-Flag: Indicates that the computation has fallen back to a lower level of disjointness than requested when set

- I-Flag: Indicates that the computation has fallen back to the best path (e.g. IGP path) and disjointness has not been achieved when set

- X-Flag : Indicates that the disjointness constraint could not be achieved and hence path has been invalidated when set

* RESERVED: 2 octets. MUST be set to 0 by the originator and MUST be ignored by a receiver.

* Disjointness Group Identifier: 4-octet value that is the group identifier for a set of disjoint paths

6.6.5. SR Bidirectional Group Constraint Sub-TLV

The SR Bidirectional Group Constraint sub-TLV is an optional sub-TLV of the SR CP Constraints TLV that is used to carry the bidirectional constraint associated with the candidate path. The bidirectional relationship between two SR Policy Candidate Paths is expressed by associating them with the same bidirectional group identifier and then specifying the type of bidirectional routing required between their paths. Only a single instance of this sub-TLV is advertised for a given CP. If multiple instances are present, then the first one is considered valid and the rest are ignored.

The 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 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Type            |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Flags            |           RESERVED            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Bidirectional Group Identifier               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

* Type: TBD

* Length: 8 octets

* Flags: two octets to indicate the bidirectional path setup information as specified in the form of flags. The following flags are defined and the other bits MUST be cleared by the originator and MUST be ignored by a receiver.

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

where:

  - R-Flag: Indicates that this CP of the SR Policy forms the reverse path when set and otherwise it is the forward path when clear

  - C-Flag: Indicates that the bidirectional path is co-routed when set

* RESERVED: 2 octets. MUST be set to 0 by the originator and MUST be ignored by a receiver.

* Bidirectional Group Identifier: 4-octet value that is the group identifier for a set of bidirectional paths

6.6.6. SR Metric Constraint Sub-TLV

The SR Metric Constraint sub-TLV is an optional sub-TLV of the SR CP Constraints TLV that is used to report the optimization metric of the CP. For a dynamic path computation, it is used to report the optimization metric used along with its parameters. For an explicit path, this sub-TLV MAY be used to report the metric margin or bound to be used for validation (i.e., the path is invalidated if the metric is beyond specified values). Multiple instances of this sub-TLV may be used to report different metric type uses.

The 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 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Metric Type  |     Flags     |            RESERVED           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Metric Margin                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Metric Bound                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

* Type: TBD

* Length: 12 octets

* Metric Type: 1-octet field which identifies the type of the metric being used. The metric type code points are listed in Section 9.8 of this document.

* 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 MUST be cleared by the originator and MUST be ignored by a receiver.

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|O|M|A|B|       |
+-+-+-+-+-+-+-+-+
```

  where:

  - O-Flag: Indicates that this is the optimization metric being reported for a dynamic CP when set. This bit MUST NOT be set in more than one instance of this TLV for a given CP advertisement.

  - M-Flag: Indicates that the metric margin allowed 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.

* RESERVED: 2 octets. MUST be set to 0 by the originator and MUST be ignored by a receiver.

* Metric Margin: 4-octet value which indicates the metric margin when the 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.

* Metric Bound: 4-octet value which indicates the maximum metric that is allowed when the B-flag is set. If the computed path metric crosses the specified bound value then the path is considered invalid.

6.7. SR Segment List TLV

The SR Segment List TLV is used to report a single SID-List of a CP. Multiple instances of this TLV may be used to report multiple SID-Lists of a CP.

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 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Flags             |           RESERVED            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             MTID              |  Algorithm    |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Weight (4 octets)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   sub-TLVs (variable)                                        //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

* Type: 1205

* Length: variable

* 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 [RFC9256]. Other bits MUST be cleared by the originator and MUST be ignored by a 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 a 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.

* RESERVED: 2 octets. MUST be set to 0 by the originator and MUST be ignored by a receiver.

* MTID: 2 octets that indicates the multi-topology identifier of the IGP topology that is to be used when the T-flag is set.

* Algorithm: 1 octet that indicates the algorithm of the SIDs used in the SID-List when the A-flag is set.

* RESERVED: 1 octet. MUST be set to 0 by the originator and MUST be ignored by a receiver.

* Weight: 4-octet field that indicates the weight associated with the SID-List for weighted load-balancing. Refer to section 2.2 and 2.11 of [RFC9256].

* 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 Sub-TLV

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 an SR Policy candidate path. It is a sub-TLV of the SR Segment List TLV and it 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Segment Type  |   RESERVED    |             Flags             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    SID (4 or 16 octets)                      //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//               Segment Descriptor (variable)                 //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//   Sub-TLVs (variable)                                       //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

* Type: 1206

* Length: variable

* Segment Type: 1 octet which indicates the type of segment (refer Section 6.8.1 for details)

* RESERVED: 1 octet. MUST be set to 0 by the originator and MUST be ignored by a receiver.

* 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 [RFC9256]. Other bits MUST be cleared by the originator and MUST be ignored by a 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.

* SID: 4 octets 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 MUST be set to 0 by the originator and MUST be ignored by a receiver.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Label                        | TC  |S|       TTL     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

* Segment Descriptor: variable size Segment descriptor based on the type of segment (refer to 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-bgpls-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.

6.8.1. Segment Descriptors

Section 4 of [RFC9256] 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 and their mapping to the respective segment types defined in [RFC9256]:

Type	Segment Description
1	(Type A) SR-MPLS Label
2	(Type B) SRv6 SID as IPv6 address
3	(Type C) SR-MPLS Prefix SID as IPv4 Node Address
4	(Type D) SR-MPLS Prefix SID as IPv6 Node Global Address
5	(Type E) SR-MPLS Adjacency SID as IPv4 Node Address & Local Interface ID
6	(Type F) SR-MPLS Adjacency SID as IPv4 Local & Remote Interface Addresses
7	(Type G) SR-MPLS Adjacency SID as pair of IPv6 Global Address & Interface ID for Local & Remote nodes
8	(Type H) SR-MPLS Adjacency SID as pair of IPv6 Global Addresses for the Local & Remote Interface
9	(Type I) SRv6 END SID as IPv6 Node Global Address
10	(Type J) SRv6 END.X SID as pair of IPv6 Global Address & Interface ID for Local & Remote nodes
11	(Type K) SRv6 END.X SID as pair of IPv6 Global Addresses for the Local & Remote Interface

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 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:

```
 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
+-+-+-+-+-+-+-+-+
|   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.

6.8.1.3. 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
 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
+-+-+-+-+-+-+-+-+
|   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
 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
+-+-+-+-+-+-+-+-+
|   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

6.8.1.5. Type 5: SR-MPLS Adjacency SID for IPv4 with an 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
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 IPv4 Node Address (4 octets)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Local Interface ID (4 octets)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

* IPv4 Node Address: 4-octet value which carries the IPv4 address associated with the node

* 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 an 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
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 IPv4 Local Address (4 octets)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                IPv4 Remote Address (4 octets)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

* 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 an 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:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           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 an 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:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Global IPv6 Local Interface Address (16 octets)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Global IPv6 Remote Interface Address (16 octets)      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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
     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
    +-+-+-+-+-+-+-+-+
    |   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 an Anterface 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
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          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).

6.8.1.11. Type 11: SRv6 END.X SID as an 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 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Global IPv6 Local Interface Address (16 octets)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Global IPv6 Remote Interface Address (16 octets)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

6.9. SR Segment List Metric Sub-TLV

The SR Segment List Metric sub-TLV reports the computed metric of the specific SID-List. It is used to report the type of metric and its computed value by the computation entity (i.e., either the headend or the controller when the path is delegated) when available. More than one instance of this sub-TLV may be present in SR Segment List to report metric values of different metric types. The metric margin and bound may be optionally reported using this sub-TLV when this information is not being reported using the SR Metric Constraint sub-TLV (refer to Section 6.6.6) at the SR CP level.

It is a sub-TLV of the SR Segment List 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 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. The metric type code points are listed in Section 9.8 of this document.

* 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 MUST be cleared by the originator and MUST be ignored by a receiver.

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|M|A|B|V|       |
+-+-+-+-+-+-+-+-+
```

  where:

  - M-Flag: Indicates that the metric margin allowed for this 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.

* RESERVED: 2 octets. MUST be set to 0 by the originator and MUST be ignored by a receiver.

* Metric Margin: 4-octet value which indicates the metric margin value when the 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 accomodate for other factors such as bandwidth availability, minimal SID stack depth, and maximizing of ECMP for the SR path computed.

* Metric Bound: 4-octet value which indicates the maximum metric value that is allowed when the B-flag is set. If the computed path metric crosses the specified bound value then the path is considered invalid.

* Metric Value: 4-octet value which indicates the metric of the computed path when the V-flag is set. This value is available and reported when the computation is successful and a valid path is available.

6.10. SR Segment List Bandwidth Sub-TLV

The SR Segment List Bandwidth sub-TLV is an optional sub-TLV used to report the bandwidth allocated to the specific SID-List by the path computation entity. Only a single instance of this sub-TLV is advertised for a given Segment List. If multiple instances are present, then the first one is considered valid and the rest are ignored.

It is a sub-TLV of the SR Segment List 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 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Bandwidth                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

* Type: TBD

* Length: 4 octets

* Bandwidth: 4 octets which specify the allocated bandwidth in unit of bytes per second in IEEE floating point format.

7. Procedures

The BGP-LS advertisements for the TE policy NLRI types 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 the reporting of SR Policy Candidate Paths, 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 state and attributes are 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 and the optimization metric for the SR Policy Candidate Path are reported using the SR Candidate Path Constraints TLV and its sub-TLVs 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 at an individual SID List level.

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 section describes the code point allocation by IANA for this document.

9.1. BGP-LS NLRI-Types

IANA maintains a registry called "BGP-LS NLRI-Types" in the "Border Gateway Protocol - Link State (BGP-LS) Parameters" registry group.

The following table lists the status of code points that have been allocated by IANA and others that are pending allocation:

Type	NLRI Type	Reference
5	SR Policy Candidate Path NLRI	this document
TBD	MPLS-TE LSP NLRI	this document
TBD	MPLS Local Cross-connect NLRI	this document

9.2. BGP-LS Protocol-IDs

IANA maintains a registry called "BGP-LS Protocol-IDs" in the "Border Gateway Protocol - Link State (BGP-LS) Parameters" registry group.

The following Protocol-ID codepoints have been allocated by IANA:

Protocol-ID	NLRI information source protocol	Reference
8	RSVP-TE	this document
9	Segment Routing	this document

9.3. BGP-LS TLVs

IANA maintains a registry called "Node Anchor, Link Descriptor and Link Attribute TLVs" in the "Border Gateway Protocol - Link State (BGP-LS) Parameters" registry group.

The following table lists the status of TLV code points that have been allocated by IANA and others that are pending allocation:

Code Point	Description	Value defined in
550	Tunnel ID	this document
551	LSP ID	this document
552	IPv4/6 Tunnel Head-end address	this document
553	IPv4/6 Tunnel Tail-end address	this document
554	SR Policy CP Descriptor	this document
555	MPLS Local Cross Connect	this document
556	MPLS Cross Connect Interface	this document
557	MPLS Cross Connect FEC	this document
1200	MPLS-TE Policy State	this document
1201	SR Binding SID	this document
1202	SR CP State	this document
1203	SR CP Name	this document
1204	SR CP Constraints	this document
1205	SR Segment List	this document
1206	SR Segment	this document
1207	SR Segment List Metric	this document
1208	SR Affinity Constraint	this document
1209	SR SRLG Constraint	this document
1210	SR Bandwidth Constraint	this document
1211	SR Disjoint Group Constraint	this document
1212	SRv6 Binding SID	this document
1213	SR Policy Name	this document
TBD	SR Bidirectional Group Constraint	this document
TBD	SR Metric Constraint	this document
TBD	SR Segment List Bandwidth	this document

9.4. BGP-LS SR Policy Protocol Origin

This document requests IANA to maintain a new registry under "Border Gateway Protocol - Link State (BGP-LS) Parameters" registry group with the allocation policy of "Expert Review" [RFC8126] using the guidelines for Designated Experts as specified in [RFC9029]. 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 with the reference set to this document:

Code Point	Protocol Origin
0	Reserved (not to be used)
1	PCEP
2	BGP SR Policy
3	Configuration (CLI, YANG model via NETCONF, etc.)
4-250	Unassigned
251-255	Private Use (not to be assigned by IANA)

9.5. BGP-LS TE State Object Origin

This document requests IANA to maintain a new registry under "Border Gateway Protocol - Link State (BGP-LS) Parameters" registry group with the allocation policy of "Expert Review" [RFC8126] using the guidelines for Designated Experts as specified in [RFC9029]. 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 with the reference set to this document:

Code Point	Object Origin
0	Reserved (not to be used)
1	RSVP-TE
2	PCEP
3	Local/Static
4-250	Unassigned
251-255	Private Use (not to be assigned by IANA)

9.6. BGP-LS TE State Address Family

This document requests IANA to maintain a new registry under "Border Gateway Protocol - Link State (BGP-LS) Parameters" registry group with the allocation policy of "Expert Review" [RFC8126] using the guidelines for Designated Experts as specified in [RFC9029]. 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 with the reference set to this document:

Code Point	Address Family
0	Reserved (not to be used)
1	MPLS-IPv4
2	MPLS-IPv6
3-250	Unassigned
251-255	Private Use (not to be assigned by IANA)

9.7. BGP-LS SR Segment Descriptors

This document requests IANA to maintain a new registry under "Border Gateway Protocol - Link State (BGP-LS) Parameters" registry group with the allocation policy of "Expert Review" [RFC8126] using the guidelines for Designated Experts as specified in [RFC9029]. 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 with the reference set to this document:

Code Point	Segment Description
0	Reserved (not to be used)
1	(Type A) SR-MPLS Label
2	(Type B) SRv6 SID as IPv6 address
3	(Type C) SR-MPLS Prefix SID as IPv4 Node Address
4	(Type D) SR-MPLS Prefix SID as IPv6 Node Global Address
5	(Type E) SR-MPLS Adjacency SID as IPv4 Node Address & Local Interface ID
6	(Type F) SR-MPLS Adjacency SID as IPv4 Local & Remote Interface Addresses
7	(Type G) SR-MPLS Adjacency SID as pair of IPv6 Global Address & Interface ID for Local & Remote nodes
8	(Type H) SR-MPLS Adjacency SID as pair of IPv6 Global Addresses for the Local & Remote Interface
9	(Type I) SRv6 END SID as IPv6 Node Global Address
10	(Type J) SRv6 END.X SID as pair of IPv6 Global Address & Interface ID for Local & Remote nodes
11	(Type K) SRv6 END.X SID as pair of IPv6 Global Addresses for the Local & Remote Interface
12-255	Unassigned

9.8. BGP-LS Metric Type

This document requests IANA to maintain a new registry under "Border Gateway Protocol - Link State (BGP-LS) Parameters" registry group with the allocation policy of "Expert Review" [RFC8126] using the guidelines for Designated Experts as specified in [RFC9029]. 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 with the reference set to this document:

Code Point	Metric Type
0	IGP Metric
1	Min Unidirectional Link Delay [RFC7471]
2	TE Metric [RFC3630]
3	Hop Count (refer [RFC5440])
4	SID List Length
5-250	Unassigned
251-255	Private Use (not to be assigned by IANA)

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, Dhanendra Jain, Francois Clad, Zafar Ali, Stephane Litkowski, and Aravind Babu Mahendra Babu for their review and valuable comments.

13. References

13.1. Normative References

[I-D.ietf-idr-bgpls-srv6-ext]
Dawra, G., Filsfils, C., Talaulikar, K., Chen, M., Bernier, D., and B. Decraene, "BGP Link State Extensions for SRv6", Work in Progress, Internet-Draft, draft-ietf-idr-bgpls-srv6-ext-13, 14 January 2023, <https://www.ietf.org/archive/id/draft-ietf-idr-bgpls-srv6-ext-13.txt>.

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

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <https://www.rfc-editor.org/info/rfc2205>.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <https://www.rfc-editor.org/info/rfc3209>.

[RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <https://www.rfc-editor.org/info/rfc3473>.

[RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <https://www.rfc-editor.org/info/rfc4090>.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <https://www.rfc-editor.org/info/rfc4760>.

[RFC4872] Lang, J.P., Ed., Rekhter, Y., Ed., and D. Papadimitriou, Ed., "RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery", RFC 4872, DOI 10.17487/RFC4872, May 2007, <https://www.rfc-editor.org/info/rfc4872>.

[RFC4873] Berger, L., Bryskin, I., Papadimitriou, D., and A. Farrel, "GMPLS Segment Recovery", RFC 4873, DOI 10.17487/RFC4873, May 2007, <https://www.rfc-editor.org/info/rfc4873>.

[RFC4874] Lee, CY., Farrel, A., and S. De Cnodder, "Exclude Routes - Extension to Resource ReserVation Protocol-Traffic Engineering (RSVP-TE)", RFC 4874, DOI 10.17487/RFC4874, April 2007, <https://www.rfc-editor.org/info/rfc4874>.

[RFC5420] Farrel, A., Ed., Papadimitriou, D., Vasseur, JP., and A. Ayyangar, "Encoding of Attributes for MPLS LSP Establishment Using Resource Reservation Protocol Traffic Engineering (RSVP-TE)", RFC 5420, DOI 10.17487/RFC5420, February 2009, <https://www.rfc-editor.org/info/rfc5420>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <https://www.rfc-editor.org/info/rfc5440>.

[RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <https://www.rfc-editor.org/info/rfc7752>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/info/rfc8986>.

[RFC9029] Farrel, A., "Updates to the Allocation Policy for the Border Gateway Protocol - Link State (BGP-LS) Parameters Registries", RFC 9029, DOI 10.17487/RFC9029, June 2021, <https://www.rfc-editor.org/info/rfc9029>.

[RFC9086] Previdi, S., Talaulikar, K., Ed., Filsfils, C., Patel, K., Ray, S., and J. Dong, "Border Gateway Protocol - Link State (BGP-LS) Extensions for Segment Routing BGP Egress Peer Engineering", RFC 9086, DOI 10.17487/RFC9086, August 2021, <https://www.rfc-editor.org/info/rfc9086>.

[RFC9256] Filsfils, C., Talaulikar, K., Ed., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", RFC 9256, DOI 10.17487/RFC9256, July 2022, <https://www.rfc-editor.org/info/rfc9256>.

13.2. Informative References

[RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, DOI 10.17487/RFC2702, September 1999, <https://www.rfc-editor.org/info/rfc2702>.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <https://www.rfc-editor.org/info/rfc3630>.

[RFC4202] Kompella, K., Ed. and Y. Rekhter, Ed., "Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4202, DOI 10.17487/RFC4202, October 2005, <https://www.rfc-editor.org/info/rfc4202>.

[RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <https://www.rfc-editor.org/info/rfc4655>.

[RFC5065] Traina, P., McPherson, D., and J. Scudder, "Autonomous System Confederations for BGP", RFC 5065, DOI 10.17487/RFC5065, August 2007, <https://www.rfc-editor.org/info/rfc5065>.

[RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, DOI 10.17487/RFC6952, May 2013, <https://www.rfc-editor.org/info/rfc6952>.

[RFC7308] Osborne, E., "Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)", RFC 7308, DOI 10.17487/RFC7308, July 2014, <https://www.rfc-editor.org/info/rfc7308>.

[RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <https://www.rfc-editor.org/info/rfc7471>.

[RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, DOI 10.17487/RFC8231, September 2017, <https://www.rfc-editor.org/info/rfc8231>.

Authors’ Addresses

Stefano Previdi
Email: stefano@previdi.net

Ketan Talaulikar (editor)
Cisco Systems
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

Hannes Gredler
RtBrick Inc.
Email: hannes@rtbrick.com

Jeff Tantsura
Microsoft
Email: jefftant.ietf@gmail.com