Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 6, 2021

J. Dong
S. Zhuang
Huawei Technologies
G. Van de Velde
Nokia

November 2, 2020

BGP Extended Community for Identifying the Target Nodes
draft-dong-idr-node-target-ext-comm-03

Abstract

BGP has been used to distribute different types of routing and policy information. In some cases, the information distributed may be only intended for one or a particular group of BGP nodes in the network. Currently BGP does not have a generic mechanism of designating the target nodes of the routing information. This document defines a new type of BGP Extended Community called "Node Target". The mechanism of using the Node Target Extended Community to steer BGP route distribution to particular BGP nodes is specified.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

BGP [RFC4271] has been used to distribute different types of routing and policy information. In some cases, the information distributed may be only intended for one or a particular group receiving BGP nodes in the network. One typical use case is the distribution of BGP Flow Spec [I-D.ietf-idr-rfc5575bis] [I-D.ietf-idr-flow-spec-v6] rules only to a particular group of BGP nodes. Such a targeting mechanism is considered useful that it can save the resources on nodes which do not need that information.

Currently BGP does not have a generic mechanism of designating the set of nodes to which the information is to be distributed. Route Target (RT) as defined in [RFC4364] was designed for the matching of VPN routes into the target VPN Routing and Forwarding tables (VRFs) on PE nodes. Although [I-D.ietf-idr-segment-routing-te-policy] introduces the mechanism of steering the SR policy information to the target head end node based on RT, it is only defined for the SR

Policy Address Family. Although it is possible to reuse RTs to control the distribution of non-VPN information to one or a group of receiving nodes, such mechanism is not applicable when the information to be distributed is VPN-specific and is advertised with a set of RTs for the VRF matching. In that case, the matching of any of the VPN RTs in the Update will result in the information eligible for installation, regardless of whether the RTs representing the target nodes are matched or not. Thus a mechanism which is independent from the control of VPN route to VRF distribution is needed.

Another possible approach is to configure, on each router, a community and the corresponding policies to match the community to determine whether to accept the received routes. Such mechanism relies on manual configuration thus is considered error-prone. It is preferable by some operators that an automatic approach can be provided, which would make the operation much easier.

This document defines a new type of BGP Extended Community called "Node Target". The mechanism of using the Node Target extended community to steer BGP route distribution to particular BGP nodes is also specified.

2. Node Target Extended Communities

This section defines a new BGP Extended Community [RFC4360] called "Node Target Extended Community". It can be a transitive extended community with the high-order octect of the type set to 0x01, or a non-transitive extended community with the high-order octect type set to 0x41. The sub-type of the Node Target Extended Community is TBA.

The format of Node Target Extended Community is shown in Figure 1.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  0x01 or 0x41 | Sub-Type(TBA) |    Target BGP Identifier      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Target BGP Identifier (cont.) |           Reserved            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1. Node Target extended community

Where:

Target BGP Identifier (4 octets): The BGP Identifier of a target node. It is a 4-octet, unsigned, non-zero integer as defined in [RFC6286].

Reserved field (2 octets): Reserved for future use, MUST be set to zero on transmission and ignored on receipt.

One or more Node Target extended communities MAY be carried in an Update message to designate a group of target BGP nodes.

3. Procedures

In this section, the mechanism for intra-domain scenario is described, the mechanism for inter-domain scenario is for further study. The domain here refers to an administrative domain, which may consists of one or multiple ASes managed by a single operator.

When a network controller or BGP speaker plans to advertise some BGP routing or policy information only to one or a group of BGP nodes in the network, it MUST put the BGP Identifier of each target node into the Node Target extended communities, and attach the Node Target extended communities to the routes or policies to be advertised.

When a BGP speaker receives a BGP Update which contains one or more Node Target extended communities, it MUST check the target BGP Identifiers carried in the Node Target extended communities of the Update.

- If the target BGP Identifier in any of the Node Target extended community matches with the local BGP Identifier, this node is one of the target nodes of the Update, the information in the Update is eligible to be kept and installed on this node.

  * If this node is a Route Reflector, and in the Update there is one or more Node Target extended communities which contains non-local BGP Identifiers, information in the Update are eligible be reflected to its peers according to the rules defined in [RFC4456]. The RR may check the BGP Identifiers of its peers to determine the set of peers which are the target nodes of the Update, and only reflect the information in the Update to the matched BGP peers.

  * If this node is an Autonomous System Border Router (ASBR), and the BGP Identifiers of one or more of its EBGP peers match with the Node Target extended communities in the Update, information in the Update is eligible to be advertised to the matched EBGP peers.

- If the target BGP Identifier in any of the Node target extended community does not match with the local BGP Identifier, this node is not the target node of Update, the information in the Update is not eligible to be installed on this node.

- If this node is a Route Reflector, information in the Update is eligible to be reflected to its peers according to the rules defined in [RFC4456]. The RR may check the BGP Identifiers of its peers to determine the set of peers which are the target nodes of the Update, and only reflect the information in the Update to the matched BGP peers.

4. Compatibility Considerations

The Node Target extended community introduced in this document can be deployed incrementally in the network. For BGP speakers which understand the Node Target extended community, it is used to determine whether the nodes are the target nodes of the Update. For BGP speakers which do not understand the Node Target extended community, it will be ignored and the information in the Update will be processed and advertised based on normal BGP procedure. Although this could ensure that the target nodes can always obtain the information needed, this may result in unnecessary state maintained on legacy BGP speakers. And if the information advertised is the Flow Spec rules, the legacy BGP speakers may install unnecessary flowspec rules, this may have impact on traffic which matches such rules, thus may result in unexpected traffic steering or filtering behaviors on such nodes. This may be mitigated by setting appropriate routing policies on the legacy BGP nodes.

5. IANA Considerations

This document requests that IANA assigns one new sub-type for "Node Target Extended Community" from the "Transitive IPv4-Address-Specific Extended Community" registry of the "BGP Eextended Communities" registry.

This document requests that IANA assigns the same sub-type for "Node Target Extended Community" from the "Non-Transitive IPv4-Address-Specific Extended Community" registry of the "BGP Eextended Communities" registry.

6. Security Considerations

This document does not change the security properties of BGP.

7. Contributors

Haibo Wang
Email: rainsword.wang@huawei.com

8. Acknowledgements

The authors would like to thank Zhenbin Li, Ercin Torun, Jeff Haas and Robert Raszuk for the review and discussion of this document.

9. References

9.1. Normative References

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

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, DOI 10.17487/RFC4360, February 2006, <https://www.rfc-editor.org/info/rfc4360>.

[RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <https://www.rfc-editor.org/info/rfc4456>.

9.2. Informative References

[I-D.ietf-idr-flow-spec-v6]
Loibl, C., Raszuk, R., and S. Hares, "Dissemination of Flow Specification Rules for IPv6", draft-ietf-idr-flow-spec-v6-18 (work in progress), November 2020.

[I-D.ietf-idr-rfc5575bis]
Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", draft-ietf-idr-rfc5575bis-27 (work in progress), October 2020.

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-09 (work in progress), May 2020.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <https://www.rfc-editor.org/info/rfc5575>.

[RFC6286] Chen, E. and J. Yuan, "Autonomous-System-Wide Unique BGP Identifier for BGP-4", RFC 6286, DOI 10.17487/RFC6286, June 2011, <https://www.rfc-editor.org/info/rfc6286>.

Authors' Addresses

Jie Dong
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: jie.dong@huawei.com

Shunwan Zhuang
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: zhuangshunwan@huawei.com

Gunter Van de Velde
Nokia
Antwerp
BE

Email: gunter.van_de_velde@nokia.com

Network Working Group
Internet Draft
Intended status: Standard
Expires: May 2, 2021

L. Dunbar
Futurewei
K. Majumdar
CommScope
H. Wang
Huawei

November 2, 2020

BGP NLRI App Meta Data for 5G Edge Computing Service
draft-dunbar-idr-5g-edge-compute-app-meta-data-01

Abstract

This draft describes a new BGP Network Layer Reachability Information (BGP NLRI) Path Attribute, AppMetaData, that can distribute the 5G Edge Computing App running status and environment, so that other routers in the 5G Local Data Network can make intelligent decision on optimized forwarding of flows from UEs. The goal is to improve latency and performance for 5G Edge Computing services.

The extension enables a feature, called soft anchoring, which makes one Edge Computing Server at one specific location to be more preferred than others for the same application to receive packets from a specific source (UE).

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes a new BGP Network Layer Reachability Information (BGP NLRI) Path Attribute, AppMetaData, that can distribute the 5G Edge Computing App running status and environment, so that other routers in the 5G Local Data Network can make intelligent decision on optimized forwarding of flows from UEs. The goal is to improve latency and performance for 5G Edge Computing services.

1.1. 5G Edge Computing Background

As described in [5G-EC-Metrics], one Application can have multiple Application Servers hosted in different Edge Computing data centers that are close in proximity. Those Edge Computing (mini) data centers are usually very close to, or co-located with, 5G base stations, with the goal to minimize latency and optimize the user experience.

When a UE (User Equipment) initiates application packets using the destination address from a DNS reply or from its own cache, the packets from the UE are carried in a PDU session through 5G Core [5GC] to the 5G UPF-PSA (User Plan Function -

PDU Session Anchor). The UPF-PSA decapsulate the 5G GTP outer header and forwards the packets from the UEs to the Ingress router of the Edge Computing (EC) Local Data Network (LDN). The LDN for 5G EC, which is the IP Networks from 5GC perspective, is responsible for forwarding the packets to the intended destinations.

When the UE moves out of coverage of its current gNB (next generation Node B) (gNB1), handover procedures are initiated and the 5G SMF (Session Management Function) also selects a new UPF-PSA. The standard handover procedures described in 3GPP TS 23.501 and TS 23.502 are followed. When the handover process is complete, the UE has a new IP address and the IP point of attachment is to the new UPF-PSA. 5GC may maintain a path from the old UPF to new the UPF for a short period of time for SSC [Session and Service Continuity] mode 3 to make the handover process more seamless.

```
   +--+
   |UE|---\+---------+                 +------------------+
   +--+    |  5G     |    +---------+  |   S1: aa08::4450 |
   +--+    | Site +--++---+         +----+                |
   |UE|----|  A   |PSA| Ra|         | R1 | S2: aa08::4460 |
   +--+    |      +---+---+         +----+                |
  +---+    |         |  |           |  |   S3: aa08::4470 |
  |UE1|---/+---------+  |           |  +------------------+
  +---+                 |IP Network |       L-DN1
                        |(3GPP N6)  |
     |                  |           |  +------------------+
     | UE1              |           |  |  S1: aa08::4450  |
     | moves to         |          +----+                 |
     | Site B           |          | R3 | S2: aa08::4460  |
     v                  |          +----+                 |
                        |           |  |  S3: aa08::4470  |
                        |           |  +------------------+
                        |           |     L-DN3
   +--+                 |           |
   |UE|---\+---------+  |           |  +------------------+
   +--+    |  5G     |  |           |  |  S1: aa08::4450  |
   +--+    | Site +--++-+--+        +----+                |
   |UE|----|  B   |PSA| Rb |        | R2 | S2: aa08::4460 |
   +--+    |      +--++----+        +----+                |
   +--+    |         | +-----------+  |  S3: aa08::4470  |
   |UE|---/+---------+                +------------------+
   +--+                                    L-DN2
```

Figure 1: App Servers in different edge DCs

1.2. Problem#1: ANYCAST in 5G EC Environment

Increasingly, Anycast is used extensively by various application providers and CDNs because ANYCAST makes it possible to dynamically load balance across server locations based on network conditions.

Application Server location selection using Anycast address leverages the proximity information present in the network (routing) layer and eliminates the single point of failure and bottleneck at the DNS resolvers and application layer load balancers. Another benefit of using ANYCAST address is removing the dependency on UEs. Some UEs (or clients) might use their cached IP addresses instead of querying DNS for extended period.

But, having multiple locations of the same ANYCAST address in 5G Edge Computing environment can be problematic because all those edge computing Data Centers can be close in proximity. There might be very little difference in the routing cost to reach the Application Servers in different Edge DCs.

BGP is an integral part in the way IP Anycast usually functions. Within BGP routing there are multiple routes for the same IP address which are pointing to different locations.

This draft describes the BGP UPDATE extension to allow the App Servers Running status and environment to be included in the BGP UPDATE messages, so that other routers can select more optimal ANYCAST location based on the combination of network delay, the App Server load index, the location capacity index and the location preference.

1.3. Problem #2: Unbalanced Anycast Distribution due to UE Mobility

Another problem of using ANYCAST address for multiple Application Servers of the same application in 5G environment is that UEs' frequent moving from one 5G site to another, which can make it difficult to plan where the App Server should be hosted. When one App server is heavily utilized, other App servers of the same address close-by can be very underutilized. Since the condition can be short lived, it is difficult for the application controller to anticipate the move and adjust.

1.4. Problem 3: Application Server Relocation

When an Application Server is added to, moved, or deleted from a 5G Edge Computing Data Center, the routing protocol needs to propagate the changes to 5G PSA or the PSA adjacent routers. After the change, the cost associated with the site [5G-EC-Metrics] might change as well.

Note: for the ease of description, the Edge Application Server and Application Server are used interchangeably throughout this document.

2. Conventions used in this document

A-ER: Egress Router to an Application Server, [A-ER] is used to describe the last router that the Application Server is attached. For 5G EC environment, the A-ER can be the gateway router to a (mini) Edge Computing Data Center.

Application Server: An application server is a physical or virtual server that host the software system for the application.

Application Server Location: Represent a cluster of servers at one location serving the same Application. One application may have a Layer 7 Load balancer, whose address(es) are reachable from external IP network, in front of a set of application servers. From IP network perspective, this whole group of servers are considered as the Application server at the location.

Edge Application Server: used interchangeably with Application Server throughout this document.

EC: Edge Computing

Edge Hosting Environment: An environment providing support required for Edge Application Server's execution.

NOTE: The above terminologies are the same as those used in 3GPP TR 23.758

Edge DC:	Edge Data Center, which provides the Edge Computing Hosting Environment. It might be co-located with 5G Base Station and not only host 5G core functions, but also host frequently used Edge server instances.
gNB	next generation Node B
L-DN:	Local Data Network
PSA:	PDU Session Anchor (UPF)
SSC:	Session and Service Continuity
UE:	User Equipment
UPF:	User Plane Function

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.

3. Usage of App Meta Data for 5G Edge Computing

3.1. Overview

From IP Layer, the Application Servers are identified by their IP (ANYCAST) addresses. The 5G Edge Computing controller or management system is aware of the ANYCAST addresses of the Applications that need optimized forwarding in 5G EC environment. The 5G Edge Computing controller or management system can configure the ACLs to filter out those applications on the routers adjacent to the 5G PSA and the routers to which the Application Servers are directly attached.

The proposed solution is for the routers, i.e. A-ER, that have direct links to the Application Servers to collect various measurements about the Servers’ running status [5G-EC-Metrics] and advertise the metrics to other routers in 5G EC LDN (Local Data Network).

3.2. IP Layer Metrics to Gauge Application Behavior

[5G-EC-Metrics] describes the IP Layer Metrics that can gauge the application servers running status and environment:

- IP-Layer Metric for App Server Load Measurement:
  The Load Measurement to an App Server is a weighted combination of the number of packets/bites to the App Server and the number of packets/bytes from the App Server which are collected by the A-ER to which the App Server is directly attached.
  The A-ER is configured with an ACL that can filter out the packets for the Application Server.
- Capacity Index
  Capacity Index is used to differentiate the running environment of the application server. Some data centers can have hundreds, or thousands, of servers behind an Application Server's App Layer Load Balancer that is reachable from external world. Other data centers can have very small number of servers for the application server. "Capacity Index", which is a numeric number, is used to represent the capacity of the application server in a specific location.
- Site preference index:
  [IPv6-StickyService] describes a scenario that some sites are more preferred for handling an application server than others for flows from a specific UE.

In this document, the term "Application Server Egress Router" [A-ER] is used to describe the last router that an Application Server is attached. For 5G EC environment, the A-ER can be the gateway router to the EC DC where multiple Application servers' instance are hosted.

From IP Layer, an Application Server is identified by its IP (ANYCAST) Address. Those IP addresses are called the Application Server IDs throughout this document.

3.3. To Equalize among Multiple ANYCAST Locations

The main benefit of using ANYCAST is to leverage the network layer information to equalize the traffic among multiple Application Server locations of the same Application, which is identified by its ANYCAST addresses.

For 5G Edge Computing environment, the ingress routers to the LDN needs to be notified of the Load Index and Capacity Index of the App Servers at different EC data centers to make the intelligent decision on where to forward the traffic for the application from UEs.

[5G-EC-Metrics] describes the algorithms that can be used by the routers directly attached to the 5G PSA to compare the cost to reach the App Servers between the Site-i or Site-j:

$$\text{Cost-i}=\min\left(w *\left(\frac{\text{Load-i} * \text{CP-j}}{\text{Load-j} * \text{CP-i}}\right) + (1-w) *\left(\frac{\text{Pref-j} * \text{Delay-i}}{\text{Pref-i} * \text{Delay-j}}\right)\right)$$

Load-i: Load Index at Site-i, it is the weighted combination of the total packets or/and bytes sent to and received from the Application Server at Site-i during a fixed time period.

CP-i: capacity index at the site I, higher value means higher capacity.

Delay-i: Network latency measurement (RTT) to the A-ER that has the Application Server attached at the site-i.

Pref-i: Preference index for the site-i, higher value means higher preference.

w: Weight for load and site information, which is a value between 0 and 1. If smaller than 0.5, Network latency and the site Preference have more influence; otherwise, Server load and its capacity have more influence.

3.4. BGP Protocol Extension to advertise Load & Capacity

Goal of the protocol extension:

- Propagate the Load Measurement Index for the attached App Servers to other routers in the LDN.

- Propagate the Capacity Index &

- Propagate Site Preference Index.

The BGP extension is to add the Load Index Sub-TLV, Capacity Sub-TLV, and the Site Preference Sub-TLV in the NLRI associated with the routes.

4. The NLRI Path Attribute for App Meta Data

The App Meta Data attribute is an optional transitive BGP Path attribute to carry application specific data, such as running status, capacity and site preference. Will need IANA to assign a value as the type code of the attribute. The attribute is composed of a set of Type-Length-Value (TLV) encodings. Each TLV contains information corresponding to metrics to a specific Application Server. An App Meta Data TLV, is structured as shown in Figure 1:

```
   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
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | AppMetaData Type (2 Octets)   |        Length (2 Octets)      |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                                                               |
  |                             Value                             |
  |                                                               |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: App Meta Data TLV Value Field

AppMetaData Type (2 octets): identifies a type of Application related metadata. The field contains values from the IANA Registry "BGP AppMetaData Types". To be added.

o Length (2 octets): the total number of octets of the value field.

o Value (variable): comprised of multiple sub-TLVs.

Each sub-TLV consists of three fields: a 1-octet type, a 1-octet or 2-octet length field (depending on the type), and zero or more octets of value. A sub-TLV is structured as shown in Figure 2:

```
+--------------------------------+
| Sub-TLV Type (1 Octet)         |
+--------------------------------+
| Sub-TLV Length (1 or 2 Octets) |
+--------------------------------+
| Sub-TLV Value (Variable)       |
+--------------------------------+
```

Figure 3: App Metadata Sub-TLV Value Field

o Sub-TLV Type (1 octet): each sub-TLV type defines a certain property about the AppMetaData TLV that contains this sub-TLV. The field contains values from the IANA Registry "BGP AppMetaData Attribute Sub-TLVs".

o Sub-TLV Length (1 or 2 octets): the total number of octets of the sub-TLV value field. The Sub-TLV Length field contains 1 octet if the Sub-TLV Type field contains a value in the range from 0-127. The Sub-TLV Length field contains two octets if the Sub-TLV Type field contains a value in the range from 128-255.

o Sub-TLV Value (variable): encodings of the value field depend on the sub-TLV type as enumerated above. The following sub-sections define the encoding in detail.

4.1. Load Measurement sub-TLV format

Two types of Load Measurement Sub-TLVs are specified. One is to carry the aggregated cost Index based on weighted combination of the collected measurements; another one is to carry the raw measurements of packets/bytes to/from the App Server address. The raw measurement is useful when the egress routers cannot be configured with a consistent algorithm to compute the aggregated load index and the raw measurements are needed by a central analytic system.

```
 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 (TBD2)           |               Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Measurement Period                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Aggregated Load Index to reach the App Server       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Aggregated Load Index Sub-TLV

Load Measurement sub-TLV has the following format:

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type (TBD3)         |               Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Measurement Period                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           total number of packets to the AppServer            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           total number of packets from the AppServer          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           total number of bytes to the AppServer              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           total number of bytes from the AppServer            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Raw Load Measurement Sub-TLV

Type =TBD2: Aggregated Load Measurement Index derived from the Weighted combination of bytes/packets sent to/received from the App server:

Index=w1*ToPackets+w2*FromPackes+w3*ToBytes+w4*FromBytes

Where w1+ w2+ w3+ w4 = 1 and 0< wi <1;

Type= TBD3: Raw measurements of packets/bytes to/from the App Server address;

Measure Period: BGP Update period or user specified period

4.2. Capacity Index sub-TLV format

The Capacity Index sub-TLV has the following format:

```
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |           Type (TBD4)         |                Length         |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                   Capacity Index                              |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Note: "Capacity Index" can be more stable for each site. If those values are configured to nodes, they might not need to be included in every BGP UPDATE.

4.3. The Site Preference Index sub-TLV format

The site Preference Index is used to achieve Soft Anchoring [Section 5] an application flow from a UE to a specific location when the UE moves from one 5G site to another.

The Preference Index sub-TLV has the following format:

```
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |           Type (TBD5)         |                Length         |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                   Preference Index                            |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Note: "Site Preference Index" can be more stable for each site. If those values are configured to nodes, they might not need to be included in every BGP UPDATE.

5. Soft Anchoring of an ANYCAST Flow

"Sticky Service" in the 3GPP Edge Computing specification (3GPP TR 23.748) requires a UE to a specific ANYCAST location when the UE moves from one 5G Site to another.

"Soft Anchoring" is referring to forwarding the Application flow from the UE to the a preferred location for the ANYCAST address, when the preferred location is in good condition. But if there is any failure at the preferred location, the Application flow from the UE need to be forwarded to another location that host the same application.

This section describes a solution that can softly anchor an application flow from a UE to a preferred location.

Lets' assume one application "App.net" is instantiated on four servers that are attached to four different routers R1, R2, R3, and R4 respectively. It is desired for packets to the "App.net" from UE-1 to stick with one server, say the App Server attached to R1, even when the UE moves from one 5G site to another. When there is failure at R1 or the Application Server attached to R1, the packets of the flow "App.net" from UE-1 need to be forwarded to the Application Server attached to R2, R3, or R4.

We call this kind of sticky service "Soft Anchoring", meaning that anchoring to the site of R1 is preferred, but other sites can be chosen when the preferred site encounters failure.

Here is details of this solution:

- Assign a group of ANYCAST addresses to one application. For example, "App.net" is assigned with 4 ANYCAST addresses, L1, L2, L3, and L4. L1/L2/L3/L4 represents the location preferred ANYCAST addresses.
- For the App.net Server attached to a router, the router has four Stub links to the same Server, L1, L2, L3, and L4 respectively. The cost to L1, L2, L3 and L4 is assigned differently for different routers. For example,
    - o When attached to R1, the L1 has the lowest cost, say 10, when attached to R2, R3, and R4, the L1 can have higher cost, say 30.

  - o ANYCAST L2 has the lowest cost when attached to R2, higher cost when attached to R1, R3, R4 respectively.
  - o ANYCAST L3 has the lowest cost when attached to R3, higher cost when attached to R1, R2, R4 respectively, and
  - o ANYCAST L4 has the lowest cost when attached to R4, higher cost when attached to R1, R2, R3 respectively
- When a UE queries for the "App.net" for the first time, the DNS replies the location preferred ANYCAST address, say L1, based on where the query is initiated.
- When the UE moves from one 5G site-A to Site-B, UE continues sending packets of the "App.net" to ANYCAST address L1. The routers will continue sending packets to R1 because the total cost for the App.net instance for ANYCAST L1 is lowest at R1. If any failure occurs making R1 not reachable, the packets of the "App.net" from UE-1 will be sent to R2, R3, or R4 (depending on the total cost to reach each of them).

If the Application Server supports the HTTP redirect, more optimal forwarding can be achieved.

- When a UE queries for the "App.net" for the first time, the global DNS replies the ANYCAST address G1, which has the same cost regardless where the Application Servers are attached.
- When the UE initiates the communication to G1, the packets from the UE will be sent to the Application Server that has the lowest cost, say the Server attached to R1. The Application server is instructed with HTTPs Redirect to respond back a location specific URL, say App.net-Loc1. The client on the UE will query the DNS for App.net-Loc1 and get the response of ANYCAST L1. The subsequent packets from the UE-1 for App.net are sent to L1.

6. Manageability Considerations

To be added.

7. Security Considerations

To be added.

8. IANA Considerations

To be added.

9. References

9.1. Normative References

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

[RFC4364] E. rosen, Y. Rekhter, "BGP/MPLS IP Virtual Private networks (VPNs)", Feb 2006.

[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>.

[RFC8200] s. Deering R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", July 2017

9.2. Informative References

[3GPP-EdgeComputing] 3GPP TR 23.748, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on enhancement of support for Edge Computing in 5G Core network (5GC)", Release 17 work in progress, Aug 2020.

[5G-EC-Metrics] L. Dunbar, H. Song, J. Kaippallimalil, "IP Layer Metrics for 5G Edge Computing Service", draft-dunbar-ippm-5g-edge-compute-ip-layer-metrics-00, work-in-progress, Oct 2020.

[5G-StickyService] L. Dunbar, J. Kaippallimalil, "IPv6 Solution for 5G Edge Computing Sticky Service", draft-dunbar-6man-5g-ec-sticky-service-00, work-in-progress, Oct 2020.

[RFC5521] P. Mohapatra, E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", April 2009.

[BGP-SDWAN-Port] L. Dunbar, H. Wang, W. Hao, "BGP Extension for SDWAN Overlay Networks", draft-dunbar-idr-bgp-sdwan-overlay-ext-03, work-in-progress, Nov 2018.

[SDWAN-EDGE-Discovery] L. Dunbar, S. Hares, R. Raszuk, K. Majumdar, "BGP UPDATE for SDWAN Edge Discovery", draft-dunbar-idr-sdwan-edge-discovery-00, work-in-progress, July 2020.

[Tunnel-Encap] E. Rosen, et al "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-10, Aug 2018.

10. Acknowledgments

Acknowledgements to Donald Eastlake for their review and contributions.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Linda Dunbar
Futurewei
Email: ldunbar@futurewei.com

Kausik Majumdar
CommScope
350 W Java Drive, Sunnyvale, CA 94089
Email: kausik.majumdar@commscope.com

Haibo Wang
Huawei
Email: rainsword.wang@huawei.com

Network Working Group
Internet Draft
Intended status: Standard
Expires: May 18, 2021

L. Dunbar
Futurewei
S. Hares
Hickory Hill Consulting
R. Raszuk
Bloomberg LP
K. Majumdar
CommScope

November 18, 2020

BGP UPDATE for SDWAN Edge Discovery
draft-dunbar-idr-sdwan-edge-discovery-01

Abstract

The document describes encoding of BGP UPDATE messages for the SDWAN edge node discovery.

In the context of this document, BGP Route Reflectors (RR) is the component of the SDWAN Controller that receives the BGP UPDATE from SDWAN edges and in turns propagates the information to the intended peers that are authorized to communicate via the SDWAN overlay network.

Status of this Memo

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on Dec 18, 2020.

Copyright Notice

Table of Contents

1. Introduction

[SDWAN-BGP-USAGE] illustrates how BGP is used as control plane for a SDWAN network. SDWAN network is an overlay network with some special properties.

The document describes a BGP UPDATE for SDWAN edge nodes to announce its properties to its RR which then propagates to the authorized peers.

2. Conventions used in this document

Cloud DC: Off-Premise Data Centers that usually host applications and workload owned by different organizations or tenants.

Controller: Used interchangeably with SDWAN controller to manage SDWAN overlay path creation/deletion and monitor the path conditions between sites.

CPE-Based VPN: Virtual Private Secure network formed among CPEs. This is to differentiate from most commonly used PE-based VPNs a la RFC 4364.

MP-NLRI: The MP_REACH_NLRI Path Attribute defined in RFC4760.

SDWAN End-point: can be the SDWAN edge node address, a WAN port address (logical or physical) of a SDWAN edge node, or a client port address.

OnPrem: On Premises data centers and branch offices

SDWAN: Software Defined Wide Area Network. In this document, "SDWAN" refers to the solutions of policy-driven transporting IP packets over multiple different underlay networks to get better WAN bandwidth management, visibility.and control.

SDWAN Instance: Same as SDWAN Segment

SDWAN Segmentation: Segmentation is the process of dividing the network into logical sub-networks. One SDWAN Segment is very much like a VPN except that SDWAN segment is over hybrid of underlay networks.

3. Framework of SDWAN Edge Discovery

3.1. The Objectives of SDWAN Edge Discovery

The objectives of SDWAN edge discovery is for a SDWAN edge node to discover its authorized peers to which its attached clients traffic need to communicate. The attributes to be propagated includes the SDWAN segmentations supported, the attached routes under specific SDWAN segmentations, and the properties of the underlay networks over which the client routes can be carried.

Some SDWAN peers are connected by both trusted VPNs and untrusted public networks. Some SDWAN peers are connected only by untrusted public networks. For the portion over untrusted networks, IPsec Security Associations (IPsec SA) have to be established and maintained. If an edge node has network ports behind the NAT, the NAT properties needs to be discovered by authorized SDWAN peers.

Just like any VPN networks, the attached client’s routes belonging to specific SDWAN segmentations can only be exchanged to the SDWAN peer nodes that are authorized to communicate.

3.2. Basic Schemes

As described in [SDWAN-BGP-USAGE], two separate BGP UPDATE messages are used for SDWAN Edge Discovery:

- UPDATE U1 for the attached client routes,
  This UPDATEs is for a SDWAN node to advertise the attached client routes to remote peers. This UPDATE will continue using the existing BGP AFI/SAFI for IP or VPN. Detailed underlay tunnel specification can be recursively resolved by using the Recursive Next Hop Resolution as specified by the section 8 of [Tunnel-Encap].

  A new Tunnel Type (SDWAN-Hybrid) needs to be added, to be used by Encapsulation Extended Community or the Tunnel-Encap Path Attribute [Tunnel-encap] to indicate mixed underlay networks.

- UPDATE U2, advertised by the Next hop address of the UPDATE U1 to propagate the properties tunnels terminated at the edge node.
  This UPDATE is for an edge node to advertise the properties of directly attached underlay networks, including the underlay network ISP information, NAT information, pre-configured IPsec SA identifiers. Also can include the detailed IPsec SA attributes, such as keys, nonce, encryption algorithms, etc.

  This UPDATE U2 is for peers to discover remote node's underlay network properties.

In the following figure: there are four types underlay paths between C-PE1 and C-PE2:

a) MPLS-in-GRE path;

b) node-based IPsec tunnel [2.2.2.2<->1.1.1.1].

c) port-based IPsec tunnel [192.0.0.1 <-> 192.10.0.10]; and

d) port-based IPsec tunnel [172.0.0.1 <-> 160.0.0.1];

```
                                       +---+
                        +--------------|RR |----------+
                       /  Untrusted    +-+-+           \
                      /                                 \
                     /                                   \
             +---------+  MPLS Path                       +-------+
     11.1.1.x| C-PE1   A1-------------------------------B1 C-PE2 |10.1.1.x
             |         |                                 |       |
     21.1.1.x|         A2(192.10.0.10)------( 192.0.0.1)B2       |20.1.1.x
             |         |                                 |       |
             | Addr    A3(160.0.0.1) --------(170.0.0.1)B3 Addr  |
             | 1.1.1.1 |                                 |2.2.2.2|
             +---------+                                 +-------+
```

Figure 1: Hybrid SDWAN

C-PE2 uses UPDATE U1 to advertise the attached client routes:

UDPATE U1:

```
      Extended community: RT for SDWAN Segmentation 1
      NLRI: AFI=? & SAFI = 1/1
        Prefix: 10.1.1.x; 20.1.1.x
        NextHop: 2.2.2.2 (C-PE2)
      Encapsulation Extended Community: tunnel-type=SDWAN-hybrid
      Color Extended Community: RED
```

The UPDATE U1 is recursively resolved to the UPDATE U2 which specifies the detailed hybrid WAN underlay Tunnels terminated at the C-PE2:

UPDATE U2:

```
      NLRI: SAFI = SDWAN
        (With Color RED encoded in the NLRI Site-Property field)
        Prefix: 2.2.2.2
        Tunnel encapsulation Path Attribute [type=SDWAN-Hybrid]
          IPSec SA for 192.0.0.1
          Tunnel-End-Point Sub-TLV [Section 3.1 of Tunnel-encap]
          IPsec SA sub-TLV [See the Section 5]
```

```
           Tunnel encapsulation Path Attribute [type=SDWAN-Hybrid]
             IPSec SA for 170.0.0.1
             Tunnel-End-Point Sub-TLV /*the address*/
             IPsec SA sub-TLV

           Tunnel Encap Attr MPLS-in-GRE [type=SDWAN-Hybrid]
             Sub-TLV for MPLS-in-GRE [Section 3.2.6 of Tunnel-encap]
```

Note: [Tunnel-Encap] Section 11 specifies that each Tunnel Encap Attribute can only have one Tunnel-End-Point sub-TLV. Therefore, two separate Tunnel Encap Attributes are needed to indicate that the client routes can be carried by either one.

3.3. Edge Node Discovery

The basic scheme of SDWAN Edge node discovery using BGP consists of:

- Upon powering up, a SDWAN edge node establish a secure tunnel (such as TLS, SSL) with the SDWAN central controller whose address is preconfigured on the edge node. The central controller will inform the edge node of its local RR. The edge node will establish a transport layer secure session with the RR (such as TLS, SSL).

- The Edge node will advertise its own properties to its designated RR via the secure transport layer tunnel. This is different from traditional BGP, where each node sends its properties (BGP UPDATE) to its neighbors, which in turn propagate to all the nodes in the network.

- The RR propagates the received information to the authorized peers.

- The authorized peers can establish the secure data channels (IPsec) and exchange more information among each other.

For a SDWAN deployment with multiple RRs, it is assumed that there are secure connections among those RRs. How secure connections being established among those RRs is the out of the scope of the current draft. The existing BGP UPDATE propagation mechanisms control the edge properties propagation among the RRs.

For some special environment where the communication to RR are highly secured, [SDN-IPsec] IKE-less can be deployed to simplify IPsec SA establishment among edge nodes.

4. BGP UPDATE to Support SDWAN Segmentation

One SDWAN network can be divided to multiple segmentations. Each SDWAN edge node may need to support multiple SDWAN segments. One client's traffic may need to be mapped to different SDWAN segmentations based on client's policy. Therefore, we need encoding to differentiate SDWAN segments. For example, in the picture below, the "Payment-Flow" (payment applications) can only be propagated to "Payment-GW". Other flows can be propagated to all other nodes. This is very similar to VPNs. But need to differentiate from traditional MPLS VPNs because a SDWAN edge may also support traditional MPLS VPNs.
[Note: SDWAN Segment ID is configured the same way as VRF, or EVI as in EVPN. For node with both MPLS and IPsec ports, the label for MPLS can be used for SDWAN Segment ID]

```
                                       +---+
                        +--------------|RR |----------+
                       /  Untrusted    +-+-+           \
                      /                                 \
                     /                                   \
             +---------+  MPLS Path                       +-------+
     11.1.1.x| C-PE1   A1-------------------------------B1 C-PE2 |10.1.1.x
             |         |                                 |       |
     21.1.1.x|         A2(192.10.0.10)------( 192.0.0.1)B2       |20.1.1.x
             |         |                                 |       |
             | Addr    A3(160.0.0.1) --------(170.0.0.1)B3 Addr  |11.2.2.x
             | 1.1.1.1 |                              /  |2.2.2.2|
             +---------+                             /   +-------+
                                                    /
                                                   /PaymentFlow
                                                  /
                                            +----+----+
                                            | payment |
                                            | Gateway |
                                            +---------+
```

Figure 2: SDWAN Segmentation

4.1. Constrained Propagation of Edge Capability

BGP has built-in mechanism to dynamically achieve the constrained distribution of edge information. RFC4684 describes the BGP RT constrained distribution. In a nutshell, a SDWAN edge sends RT Constraint (RTC) NLRI to the RR for the RR to install an outbound route filter, as shown in the figure below:

```
    RT Constraint                   RT constraint
    NLRI={SDWAN#1, SDWAN#2}         NLRI={SDWAN#1, SDWAN#3}
            ----->                 +---+      <-----------
              +--------------------|RR1|------------------+
              | Outbound Filter    +---+  Outbound Filter |
              | Permit SDWAN#1,#2        Permit SDWAN#1,#3|
              | Deny all                 Deny all         |
              |   <-------                ---------->      |
              |                                           |
        +-----+---+  MPLS Path                      +-----+-+
11.1.1.x| C-PE1   A1-------------------------------B1 C-PE2 |10.1.1.x
        |         |                                 |       |
21.1.1.x|         A2(192.10.0.10)------( 192.0.0.1)B2       |20.1.1.x
        |         |                                 |       |
        | Addr    A3(160.0.0.1) --------(170.0.0.1)B3 Addr  |
        | 1.1.1.1 |                                 |2.2.2.2|
        +---------+                                 +-------+
SDWAN Instance #1                                   SDWAN Instance #1
SDWAN Instance #2                                   SDWAN Instance #3
```

Figure 3: Constraint propagation of Edge Property

However, as SDWAN overlay network span across untrusted networks, RR can’t trust the RT Constraint (RTC) NLRI BGP UPDATE from any nodes. Polices must be configured on RR to filter out unauthorized nodes to be registered as interested in certain SDWAN segments. RR can only process the RTC NLRI from authorized peers for a SDWAN segment.

It is out of the scope of this document on how RR is configured with the policies to filter out unauthorized nodes for specific SDWAN segments.

When the RR receives BGP UPDATE from an edge node, it propagates the received UPDATE message to the nodes that are in the Outbound Route filter for the specific SDWAN segment.

4.2. SDWAN Segmentation for Control Plane

SDWAN Instances is represented by the SDWAN Target ID in the BGP Extended Community.

Same as Route Target for VPN, a different Type is used to differentiate SDWAN segments from MPLS VPN instances. This is especially useful when a CPE supports both MPLS VPN and SDWAN Segmentation (instances).

Encoding:

```
 RFC4360: Extended Community for SDWAN Route Target
  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 high     | Type low(*)   |                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+          Value                |
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 0 1 2 3 4 5 6 7
 +-+-+-+-+-+-+-+-+
 |I|T| 6-bit val |
 +-+-+-+-+-+-+-+-+
```

The high-order octet of the Type Field
T bit =0 (transitive) when SDWAN edge sends to its RR which then propagates to remote peers based on outbound filters.

RFC4760 states that Route Target community is transitive
For SDWAN, an edge receiving the SDWAN Update shouldn’t forward it to other nodes.
T bit =1 (non-transitive) when RR propagates the UPDATE to SDWAN EDGE

[IANA Consideration:

Following the encoding scheme specified by RFC7153, need IANA to assign the following values for the "Type High" Octet:

- Transitive (when edge announce the advertisement to its RR): Ox0A, which is the number after 0x08 for Flow Spec Redirect.
- Non Transitive (when RR send to remote edges): Ox4A

Request a new value of the low- order octet of the Type field for this community (different from the VPN Route Target 0x02)?
]

4.3. SDWAN Segment Identifier in Data Plane

From data plane perspective, packets from different SDWAN network instances (or segmentations) need to have their corresponding SDWAN instance identifier encoded in the header.

For a SDWAN edge node which can be reached by both MPLS and IPsec path, the client packets reached by MPLS network will be encoded with the MPLS Labels based on the scheme specified by RFC8277.

For GRE Encapsulation within IPsec tunnel, the GRE key field can be used to carry the SDWAN Instance ID. For NVO (VxLAN, GENEVE, etc.) encapsulation within the IPsec tunnel, Virtual Network Identifier (VNI) field is used to carry the SDWAN Instance ID.

[Note: the SDWAN Instance ID is same as EVI in EVPN, or VNI if VxLAN is used].

5. Hybrid Underlay

5.1. SDWAN-Hybrid Tunnel Encoding

A new Tunnel-Type=SDWAN-Hybrid (code point to be assigned by IANA)is introduced to indicate hybrid underlay networks.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Tunnel-Type(=SDWAN-Hybrid )   | Length (2 Octets)             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                             Value                             |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SDWAN Hybrid Underlay network Sub-TLV Value Field

Since IPsec SA has a lot of attributes, such as public keys, nonce, encryption algorithms etc., the IPsec Tunnel Identifier [ID] can be used in the SDWAN-Hybrid Value field instead of listing all IPsec SA attributes. Using IPsec Tunnel ID can greatly reduce the size of BGP UPDATE messages. Another added benefit of using IPsec Tunnel ID is that the IPsec SA attributes, or rekeying process can be advertised independently.

There are two Sub-TLVs to represent the IPsec IDs under the SDWAN-Hybrid tunnel type: IPsec-SA-ID and IPsec-SA-Group.

Editor’s note: The IPSEC-SA-Group was designed to provide better scaling for multiple IPsec SA terminated at one endpoint. One end point can have multiple IPsec SAs, one SA can encrypt client data to CPE1 and another one for CPE2.

IPsec-SA-ID Sub-TLV

```
    0                   1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    IPsec SA Identifier        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The IPsec SA identifier (2 Octet) is for cross reference the IPsec SA attributes being pre-configured or advertised by another UPDATE [Section 7].

If the client traffic needs to be encapsulated in a specific type within the IPsec ESP Tunnel, such as GRE or VxLAN, etc., the corresponding Tunnel-Encap Sub-TLV needs to be appended right after the IPsec-SA-ID Sub-TLV.

Editor Note: 4 octets can be considered as well for IPsec-SA-ID.

IPsec-SA-Group Sub-TLV:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |        InnerEncapType         |            Length             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  IPsec SA Identifier #1       |   IPsec SA Identifier #2      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                               |IPsec SA Identifier #n         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                    IPsec-SA-Group Sub-TLV
```

IPsec-SA-Group Sub-TLV is for the scenario that multiple IPsec SAs have the same inner encapsulation. Multiple IPsec SA IDs are included in the IPsec-ID-Group Sub-TLV. If different inner encapsulation is desired within IPsec tunnels, then multiple IPsec-SA-Group Sub-TLVs can be included within one Tunnel Encap Path Attribute.

InnerEncapType (2 octet) indicates the encapsulation type for the payload within the IPsec ESP Tunnel. The Inner Encap Type value will take the value specified by the IANA Consideration Section (12.5) of [Tunnel-Encap]:

- types 8 (VXLAN), 9 (NVGRE), 11 (MPLS-in-GRE), and 12 (VXLAN-GPE) in the "BGP Tunnel Encapsulation Tunnel Types" registry.
- types 1 (L2TPv3), 2 (GRE), and 7 (IP in IP) in the "BGP Tunnel Encapsulation Tunnel Types" registry.

For each of the Tunnel Types specified, the detailed encapsulation value field as specified by Section 3.2 of [Tunnel-Encap] is appended right after the IPsec Sub-TLV.

The Tunnel Ending Point Sub-TLV specified by the Section 3.1 of [Tunnel-Encap] has to be attached to identify the IPsec Tunnel terminating address.
There can be multiple IPsec tunnels terminating at one WAN port or at one node, e.g. one tunnel for going to destination "A", another one for going to destination "B". Use SDWAN for retail industry as an example, it is necessary for all shops at any location to only

exchange Payment System traffic with the Payment Gateway, while other traffic can be exchanged with any nodes.
Therefore, there could be multiple IPsec Sub-TLVs bound with one Tunnel Ending Point Sub-TLV.

However, it is quite common in SDWAN deployment that all IPsec attributes from one node or one port are the same for all destinations. In that case, IPsec SA ID is set to 0 and the terminating address can be used to cross reference the IPsec SA attributes which are advertised by the Underlay Network Property advertisement UPDATE.

5.2. Encoding Example

5.2.1. Multiple IPsec SAs Sharing One Tunnel End Point

The encoding example is for the following scenario:

- there are three IPsec SAs terminating at the same WAN Port address (or the same node address)
- Two of the IPsec SAs use GRE (value =2) as Inner Encapsulation within the IPsec Tunnel
- One of the IPsec SA uses VxLAN (value = 8) as the Inner Encapsulation within its IPsec Tunnel.

Here is the encoding for the scenario:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Tunnel-Type =SDWAN-Hybrid     |       Length =                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            Tunnel-end-Point Sub-TLV                           |
   ~                                                               ~

   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |IPsec-SA-group:InnerEncapType=2|       Length = 4              |
   +-------------------------------+-------------------------------+
   |  IPsec SA Identifier = 1      |    IPsec SA Identifier = 2    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                   GRE-KEY (4 Octets)                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |IPsec-SA-ID:  =3               |        Reserved               |
   +-------------------------------+-------------------------------+
   ~                      VxLAN Sub-TLV                            ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Length of the Tunnel-Type = SDDWAN-Hybrid is the sum of the following:

- Tunnel-end-point sub-TLV total length
- the IPsec-SA-Group Sub-TLV length + 4 (the two octets for InnerEncapType + the two octets for the Length field)
- GRE-Key Length (4)
- The IPsec-SA-ID Sub-TLV length: 4
- The VxLAN sub-TLV total length

5.2.2. Multiple IPsec SAs with different Tunnel End Points

If IPsec SAs are terminating at different addresses, then multiple Tunnel Encap Attributes have to be included.

The encoding example for the Figure 1:

- there is one IPsec SA terminating at the WAN Port address 192.0.0.1; and another IPsec SA terminating at WAN Port 170.0.0.1;
- Both IPsec SAs use GRE (value =2) as Inner Encapsulation within the IPsec Tunnel

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Tunnel-Type =SDWAN-Hybrid     |       Length =                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |            Tunnel-end-Point Sub-TLV                           |
    ~            for  192.0.0.1                                     ~
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  IPsec SA Identifier = 1      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ~                   GRE Sub-TLV                                 ~
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Tunnel-Type =SDWAN-Hybrid     |       Length =                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |            Tunnel-end-Point Sub-TLV                           |
    ~            for  170.0.0.1                                    ~
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  IPsec SA Identifier = 1      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ~                   GRE sub-TLV                                 ~
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

6. Hybrid Underlay Detailed Attributes

6.1. SDWAN NLRI for Underlay Network Properties

For the MPLS VPN, the underlay network is controlled by the VPN service provider, therefore, there is no need for nodes to advertise any underlay properties to remote peers.

For the untrusted underlay network to which a SDWAN edge is connected, many attributes need to be advertised to remote nodes, such as:

- ISP information of the underlay network,
- NAT property
- the geolocation of the SDWAN edge
- IPsec SA attributes, such as public keys, nonce, supported encryption algorithms, etc.
- the IPsec tunnel termination address

A new SDWAN NLRI is specified within the MP_REACH_NLRI Path Attribute of RFC4760, with SAFI=SDWAN (code = 74):

```
     +-------------------+
     |   NLRI Length     | 1 octet
     +-------------------+
     |   Site-Type       | 1 Octet
     +-------------------+
     |   Port-Local-ID   | 4 octets
     +-------------------+
     |  SDWAN-Color      | 4 octets
     +-------------------+
     |  SDWAN-Node-ID    | 4 or 16 octets
     +-------------------+
```

where:

- NLRI Length: 1 octet of length expressed in bits as defined in [RFC4760].
- Site Type: 1 octet value. The SDWAN Site Type defines the different types of Site IDs to be used in the deployment. The draft defines the following types:
  Site-Type = 1: For simple deployment, such as all edge nodes under one SDWAN management system, a simple identifier is enough for the SDWAN management to map the site to its precise geolocation.
  Site-Type = 2: to indicate that the value in the site-ID is locally significant, therefore, need a Geo-Loc Sub-TLV to fully describe the accurate location of the node. This is for large SDWAN heterogeneous deployment where Site IDs has to be described by proper Geo-location of the Edge Nodes [LISP-GEOLoc].
- Port local ID: SDWAN edge node Port identifier, which can be locally significant. The detailed properties about the network connected to the port are further encoded in the Tunnel Path Attribute. If the SDWAN NLRI applies to multiple ports, this field is NULL.
- SDWAN-Color: is used to correlate with the Color-Extended-community included in the client routes UPDATE. It can also represent some common properties shared by a set of SDWAN edge

nodes, such as the property of a specific geographic location shared by a group of SDWAN edge nodes.

- SDWAN Edge Node ID: a routable address (IPv4 or IPv6) within the WAN to reach this node or port.

[Editor's note on using SDWAN SAFI for the underlay network property advertisement:

SDWAN SAFI [IANA assigned =74] is used instead of IP SAFI in the MP-NLRI [RFC4760] Path Attribute to advertise the underlay network properties to emphasize that the address in the NLRI is NOT client addresses.

If the same IP SAFI used, receiver needs to add extra logic to differentiate regular BGP MP-NLRI client routes advertisement from the SDWAN underlay network properties advertisement. The benefit of using the same IP SAFI is that the UPDATE can traverse existing routers without being dropped. Since the SDWAN underlay network UPDATE is only between SDWAN edge and its corresponding RR, there won't be any intermediated routers. Therefore, this benefit becomes not applicable.

]

6.2. Extended Port Sub-TLV

When a SDWAN edge node is connected to an underlay network via a port behind NAT devices, traditional IPsec uses IKE for NAT negotiation. The location of a NAT device can be such that:

- Only the initiator is behind a NAT device. Multiple initiators can be behind separate NAT devices. Initiators can also connect to the responder through multiple NAT devices.
- Only the responder is behind a NAT device.
- Both the initiator and the responder are behind a NAT device.

The initiator's address and/or responder's address can be dynamically assigned by an ISP or when their connection crosses a dynamic NAT device that allocates addresses from a dynamic address pool.

Because one SDWAN edge can connect to multiple peers via one underlay network, the pair-wise NAT exchange as IPsec's IKE is not efficient. In BGP Controlled SDWAN, NAT information of a WAN port is advertised to its RR in the BGP UPDATE message. It is encoded as an Extended sub-TLV that describes the NAT property if the port is behind a NAT device.

A SDWAN edge node can inquire STUN (Session Traversal of UDP Through Network Address Translation RFC 3489) Server to get the NAT property, the public IP address and the Public Port number to pass to peers.

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Port Ext Type  |  EncapExt subTLV Length      |I|O|R|R|R|R|R|R|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | NAT Type      |  Encap-Type   |Trans networkID|     RD ID     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                  Local  IP Address                            |
               32-bits for IPv4, 128-bits for Ipv6
                       ~~~~~~~~~~~~

    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                  Local  Port                                  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                Public IP                                      |
               32-bits for IPv4, 128-bits for Ipv6
                       ~~~~~~~~~~~~

    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                Public Port                                    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                ISP-Sub-TLV                                    |
    ~                                                               ~

    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

o Port Ext Type: indicate it is the Port Ext SubTLV.
o PortExt subTLV Length: the length of the subTLV.
o Flags:
    - I bit (CPE port address or Inner address scheme)

If set to 0, indicate the inner (private) address is IPv4. If set to 1, it indicates the inner address is IPv6.

- O bit (Outer address scheme): If set to 0, indicate the public (outer) address is IPv4. If set to 1, it indicates the public (outer) address is IPv6.

- R bits: reserved for future use. Must be set to 0 now.

- NAT Type.without NAT; 1:1 static NAT; Full Cone; Restricted Cone; Port Restricted Cone; Symmetric; or Unknown (i.e. no response from the STUN server).
- Encap Type.the supported encapsulation types for the port facing public network, such as IPsec+GRE, IPsec+VxLAN, IPsec without GRE, GRE (when packets don’t need encryption)
- Transport Network ID.Central Controller assign a global unique ID to each transport network.
- RD ID.Routing Domain ID.need to be global unique.
- Local IP.The local (or private) IP address of the port.
- Local Port.used by Remote SDWAN edge node for establishing IPsec to this specific port.
- Public IP.The IP address after the NAT. If NAT is not used, this field is set to NULL.
- Public Port.The Port after the NAT. If NAT is not used, this field is set to NULL.

6.3. ISP of the Underlay network Sub-TLV

The purpose of the Underlay network Sub-TLV is to carry the ISP WAN port properties with SDWAN SAFI NLRI. It would be treated as optional Sub-TLV. The BGP originator decides whether to include this Sub-TLV along with the SDWAN NLRI. If this Sub-TLV is present, it would be processed by the BGP receiver and to determine what local policies to apply for the remote end point of the Underlay tunnel.

The format of this Sub-TLV is as follows:

```
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |      Type     |     Length    |      Flag     |    Reserved   |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |Connection Type|   Port Type   |         Port Speed            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

Type: To be assigned by IANA

Length: 6 bytes.

Flag: a 1 octet value.

Reserved: 1 octet of reserved bits. It SHOULD be set to zero on transmission and MUST be ignored on receipt.

Connection Type: There are two different types of WAN Connectivity. They are listed below as:

Wired	1
WIFI	2
LTE	3
5G	4

Port Type: There are different types of ports. They are listed Below as:

Ethernet	1
Fiber Cable	2
Coax Cable	3
Cellular	4

Port Speed: The port seed is defined as 2 octet value. The values are defined as Gigabit speed.

7. IPsec Security Association Property Sub-TLVs

7.1. Controller Facilitated IPsec Tunnels for SDWAN Networks

IPsec is a common technique used to encrypt traffic traversing untrusted networks. IPSec operation between two peer nodes need to perform Internet Key Exchange (IKEv2), which can be broken down into the following steps:

- IKE_SA_INIT exchanges: This pair of messages negotiate cryptographic algorithms, exchange nonces, and do a Diffie-Hellman exchange.

- IKE_AUTH: this pair of messages authenticate the previous messages, exchange identities and certificates, and establish the first Child SA. Based on the authentication used: Pre-Shared Key, RSA certificates or EAP the number of messages exchanged in IKE_AUTH can grow.

- CREATE_CHILD_SA - This is simply used to create additional CHILD-SAs as needed

- INFORMATIONAL- During an IKEv2 SA lifetime, peers may desire to exchange some control messages related to errors or have notifications of certain events. This function is accomplished by INFORMATIONAL exchange.

In SDWAN environment, each SDWAN edge node might need to establish IPsec tunnels to multiple peers, and there can be multiple IPsec tunnels for different client traffic between any two peers. In addition, SDWAN edge nodes can be far apart. Upon powering up, a SDWAN edge might not know their authorized communication peers and might not have the policies configured for aligning traffic with their specific IPsec Tunnels. Therefore, the IPsec operation in SDWAN environment are usually facilitated by its SDWAN Controller.

[SDN-IPsec] describes two different mechanisms to achieve controller facilitated IPsec configuration: IKE case vs. IKE-less case. Under the IKE case, the Controller is in charge of provisioning the required information to IKE, the Security Policy Database (SPD) and the Security Association Database (PAD). The SDWAN peers exchange the Internet Key Exchange (IKE) protocol and manage SPD and SAD. Under the IKE-less case, the Controller will provide the required parameters to create valid entries in the SPD and the SAD into the edge nodes. Therefore, the edge node will only need to implementation IPsec encryption while automated key management functionality is moved to the Controller.

For BGP controlled SDWAN networks, since there is already a secure management tunnel established between RR and the edge nodes, all the negotiations exchanged in IKEv2 can be carried by BGP UPDATE messages to/from the Route Reflector (RR). RR will propagate the information to the intended destinations. More importantly, when an edge node needs to establish multiple IPsec tunnels to many different SDWAN edge nodes, all the management information can be multiplexed into the secure management tunnel between RR and the edge node. Therefore, there is reduced amount of work on authentication in processing in BGP Controlled SDWAN networks.

Editor’s Note:

RFC7296 specifies the IPsec SA attributes exchange among two peers to establish peer-wise IPsec SA. [Controller-IKE] specifies the structure for a controller to facilitate the exchange of the RFC7296 specified IPsec SA attributes among many nodes.

[CONTROLLER-IKE] specifies the Device Information Message (DIM) for the edge node to advertise to its controller, which includes DH public number, nonce, peer identity, an indication whether this is the initial distributed policy, and rekey counter. The originating edge node distributes the DH public value along with the other values in the DIM (using IPsec Tunnel TLV in Tunnel Encapsulation Attribute) to other remote C-PEs via the RR. Each receiving C-PE uses this DH public number and the corresponding nonce in creation of IPsec SA pair to the originating C-PE - i.e., an outbound SA and an inbound SA. The detail procedures are described in section 5.2 of [CONTROLLER-IKE].

[SECURE-VPN] proposes the BGP UPDATE Sub-TLV structure to carry the information specified by [Controller-IKE] to be propagated among peers via BGP.

To expedite the standardization process, this draft aligns with the IPsec Sub-TLVs described in the Section 6.1, 6.2 and 6.3 of [SECURE-EVPN], with some optimization.

For scalability reason, this draft advertises the IPSec SA related attributes at a different pace than client routes UPDATEs. Client Routes UPDATE only references the identifier for the prior established IPsec SAs.

The optimized IPsec SA attributes are represented by a set of Sub-TLVs:

- IPsec SA Nonce Sub-TLV
- IPsec SA Public Key Sub- TLV
- IPsec SA Proposal Sub-TLV: to indicate the number of Transform Sub-TLVs
  - o Transforms Substructure Sub-TLV

For BGP controlled SDWAN network, very often an edge node doesn't know its peer identity. Then the peer identity field can be null.

7.2. IPsec SA Nonce Sub-TLV

The Nonce Sub-TLV is based on the Base DIM sub-TLV as described the Section 6.1 of [SECURE-EVPN]. IPsec SA ID is added to the sub-TLV, which is to be referenced by the client route NLRI Tunnel Encap Path Attribute for the IPsec SA. The following fields are removed because:

- the Originator ID is carried by the NLRI,
- the Tenant ID is represented by the Route Target Extended Community, and
- the Subnet ID are carried by the BGP route UPDATE.

The format of this Sub-TLV 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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |   ID Length   |       Nonce Length            |I|   Flags     |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                             Rekey                             |
       |                            Counter                            |
       +---------------------------------------------------------------+
       |      IPsec SA ID              |        Reserved               |
       +---------------------------------------------------------------+
       |                                                               |
       ~                          Nonce Data                           ~
       |                                                               |
       +---------------------------------------------------------------+
```

IPsec SA ID - The 2 bytes IPSec SA ID could 0 or non-zero values. It is cross referenced by client route’s IPSec Tunnel Encap IPSec-SA-ID or IPSec-SA-Group Sub-TLV in Section 5. When there are multiple IPsec SAs terminated at one address, such as WAN port address or the node address, they are differentiated by the different IPsec SA IDs.

7.3. IPsec Public Key Sub-TLV

The IPsec Public Key Sub-TLV is derived from the Key Exchange Sub-TLV described in [SECURE-EVPN] with an addition of Duration filed to define the IPSec SA life span. The edge nodes would pick the shortest duration value between the SDWAN SAFI pairs.

The format of this Sub-TLV 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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |   Diffie-Hellman Group Num    |          Reserved             |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       ~                       Key Exchange Data                       ~
       |                                                               |
       +---------------------------------------------------------------+
       |                            Duration                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

7.4. IPsec SA Proposal Sub-TLV

The IPsec SA Proposal Sub-TLV is to indicate the number of Transform Sub-TLVs. This Sub-TLV aligns with the sub-TLV structure from [SECURE-VPN]

The Transform Sub-sub-TLV will following the section 3.3.2 of RFC7296.

7.5. Simplified IPsec Security Association sub-TLV

For a simple SDWAN network with edge nodes supporting only a few pre-defined encryption algorithms, a simple IPsec sub-TLV can be used to encode the pre-defined algorithms, as below:

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |IPsec-simType  |IPsecSA Length                 | Flag          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Transform     | Mode          | AH algorithms |ESP algorithms |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |         ReKey Counter (SPI)                                   |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | key1 length   |         Public Key                            ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | key2 length   |         Nonce                                 ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |        Duration                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

- o IPsec-SimType: The type value has to be between 128~255 because IPsec-SA subTLV needs 2 bytes for length to carry the needed information.
- o IPsec-SA subTLV Length (2 Byte): 25 (or more)
- o Flags: 1 octet of flags. None are defined at this stage. Flags SHOULD be set to zero on transmission and MUST be ignored on receipt.
- o Transform (1 Byte): the value can be AH, ESP, or AH+ESP.

- o IPsec Mode (1 byte): the value can be Tunnel Mode or Transport mode
- o AH algorithms (1 byte): AH authentication algorithms supported, which can be md5 | sha1 | sha2-256 | sha2-384 | sha2-512 | sm3. Each SDWAN edge node can have multiple authentication. algorithms; send to its peers to negotiate the strongest one.
- o ESP (1 byte): ESP authentication algorithms supported, which can be md5 | sha1 | sha2-256 | sha2-384 | sha2-512 | sm3. Each SDWAN edge node can have multiple authentication algorithms; send to its peers to negotiate the strongest one. Default algorithm is AES-256.
    - o When node supports multiple authentication algorithms, the initial UPDATE needs to include the "Transform Sub-TLV" described by [SECURE-EVPN] to describe all of the algorithms supported by the node.

- o Rekey Counter (Security Parameter Index)): 4 bytes
- o Public Key: IPsec public key
- o Nonce.IPsec Nonce
- o Duration: SA life span.

7.6. IPsec SA Encoding Examples

For the Figure 1 in Section 3, C-PE2 needs to advertise its IPsec SA associated attributes, such as the public keys, the nonce, the supported encryption algorithms for the IPsec tunnels terminated at 192.0.0.1, 170.1.1.1 and 2.2.2.2 respectively.

Using the IPsec Tunnel [ISP4: 160.0.0.1 <-> ISP2:170.0.0.1] as an example: C-PE1 needs to advertise the following attributes for establishing the IPsec SA:

```
  SDWAN Node ID
  SDWAN Color
  Tunnel Encap Attr (Type=SDWAN-Hybrid)
       Extended Port Sub-TLV for information about the Port
       (including ISP Sub-TLV for information about the ISP2)
       IPsec SA Nonce Sub-TLV,
       IPsec SA Public Key Sub-TLV,
       IPsec SA Sub-TLV for the supported transforms
```

```
               {Transforms Sub-TLV - Trans 2,
               Transforms Sub-TLV - Trans 3}
```

C-PE2 needs to advertise the following attributes for establishing IPsec SA:

```
     SDWAN Node ID
     SDWAN Color
     Tunnel Encap Attr (Type=SDWAN-Hybrid)
          Extended Port Sub-TLV (including ISP Sub-TLV for information
          about the ISP2)
          IPsec SA Nonce Sub-TLV,
          IPsec SA Public Key Sub-TLV,
          IPsec SA Sub-TLV for the supported transforms
               {Transforms Sub-TLV - Trans 2,
               Transforms Sub-TLV - Trans 4}
```

As both end points support Transform #2, the Transform #2 will be used for the IPsec Tunnel [ISP4: 160.0.0.1 <-> ISP2:170.0.0.1].

8. Error & Mismatch Handling

Each C-PE device advertises SDWAN SAFI Underlay NLRI to the other C-PE devices via BGP Route Reflector to establish pairwise SAs between itself and every other remote C-PEs. During the SAFI NLRI advertisement, the BGP originator would include either simple IPSec Security Association properties defined in IPSec SA Sub-TLV based on IPSec-SA-Type = 1 or full-set of IPSec Sub-TLVs including Nonce, Public Key, Proposal and number of Transform Sub-TLVs based on IPSec-SA-Type = 2.

The C-PE devices would compare the IPSec SA attributes between the local and remote WAN ports. If there is a match on the SA Attributes between the two ports, the IPSec Tunnel would be established.

The C-PE devices would not try to negotiate the base IPSec-SA parameters between the local and the remote ports in the case of simple IPSec SA exchange or the Transform sets between local and remote ports if there is a mismatch on the Transform sets in the case of full-set of IPSec SA Sub-TLVs.

As an example, using the Figure 1 in Section 3, to establish IPsec Tunnel between C-PE1 and C-PE2 WAN Ports A2 and B2 [A2: 192.10.0.10 <-> B2:192.0.0.1]:

C-PE1 needs to advertise the following attributes for establishing the IPsec SA:

```
  NH: 192.10.0.10
  SDWAN Node ID
  SDWAN-Site-ID
  Tunnel Encap Attr (Type=SDWAN)
       ISP Sub-TLV for information about the ISP3
       IPsec SA Nonce Sub-TLV,
       IPsec SA Public Key Sub-TLV,
       Proposal Sub-TLV with Num Transforms = 1
            {Transforms Sub-TLV - Trans 1}
```

C-PE2 needs to advertise the following attributes for establishing IPsec SA:

```
  NH: 192.0.0.1
  SDWAN Node ID
  SDWAN-Site-ID
  Tunnel Encap Attr (Type=SDWAN)
       ISP Sub-TLV for information about the ISP1
       IPsec SA Nonce Sub-TLV,
       IPsec SA Public Key Sub-TLV,
       Proposal Sub-TLV with Num Transforms = 1
            {Transforms Sub-TLV - Trans 2}
```

As there is no matching transform between the WAN ports A2 and B2 in C-PE1 and C-PE2 respectively, there will be no IPsec Tunnel be established.

9. Manageability Considerations

TBD - this needs to be filled out before publishing

10. Security Considerations

The document describes the encoding for SDWAN edge nodes to advertise its properties to their peers to its RR, which propagates to the intended peers via untrusted networks.

The secure propagation is achieved by secure channels, such as TLS, SSL, or IPsec, between the SDWAN edge nodes and the local controller RR.

[More details need to be filled in here]

11. IANA Considerations

This document requires the following IANA actions.

- o SDWAN Overlay SAFI = 74 assigned by IANA
- o SDWAN Route Type

12. References

12.1. Normative References

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

12.2. Informative References

[RFC8192] S. Hares, et al, "Interface to Network Security Functions (I2NSF) Problem Statement and Use Cases", July 2017

[RFC5521] P. Mohapatra, E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", April 2009.

[CONTROLLER-IKE] D. Carrel, et al, "IPsec Key Exchange using a Controller", draft-carrel-ipsecme-controller-ike-01, work-in-progress.

[LISP-GEOLOC] D. Farinacci, "LISP Geo-Coordinate Use-Case", draft-farinacci-lisp-geo-09, April 2020.

[SDN-IPSEC] R. Lopez, G. Millan, "SDN-based IPsec Flow Protection", draft-ietf-i2nsf-sdn-ipsec-flow-protection-07, Aug 2019.

[SECURE-EVPN] A. Sajassi, et al, "Secure EVPN", draft-sajassi-bess-secure-evpn-02, July 2019.

[Tunnel-Encap]E. Rosen, et al, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-09, Feb 2018.

[VPN-over-Internet] E. Rosen, "Provide Secure Layer L3VPNs over Public Infrastructure", draft-rosen-bess-secure-l3vpn-00, work-in-progress, July 2018

[DMVPN] Dynamic Multi-point VPN: https://www.cisco.com/c/en/us/products/security/dynamic-multipoint-vpn-dmvpn/index.html

[DSVPN] Dynamic Smart VPN: http://forum.huawei.com/enterprise/en/thread-390771-1-1.html

[ITU-T-X1036] ITU-T Recommendation X.1036, "Framework for creation, storage, distribution and enforcement of policies for network security", Nov 2007.

[Net2Cloud-Problem] L. Dunbar and A. Malis, "Seamless Interconnect Underlay to Cloud Overlay Problem Statement", draft-dm-net2cloud-problem-statement-02, June 2018

[Net2Cloud-gap] L. Dunbar, A. Malis, and C. Jacquenet, "Gap Analysis of Interconnecting Underlay with Cloud Overlay", draft-dm-net2cloud-gap-analysis-02, work-in-progress, Aug 2018.

[Tunnel-Encap] E. Rosen, et al "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-10, Aug 2018.

13. Acknowledgments

Acknowledgements to Wang Haibo, Hao Weiguo, and ShengCheng for implementation contribution; Many thanks to Jim Guichard, John Scudder, and Donald Eastlake for their review and contributions.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Linda Dunbar
Futurewei
Email: ldunbar@futurewei.com

Sue Hares
Hickory Hill Consulting
Email: shares@ndzh.com

Robert Raszuk
Email: robert@raszuk.net

Kausik Majumdar
CommScope
Email: Kausik.Majumdar@commscope.com

INTERNET-DRAFT D. Eastlake
Intended Status: Proposed Standard Futurewei Technologies
W. Hao
S. Zhuang
Z. Li
Huawei Technologies
R. Gu
China Mobil
Expires: March 29, 2021 September 30, 2020

BGP Dissemination of
Flow Specification Rules for Tunneled Traffic
draft-ietf-idr-flowspec-nvo3-10

Abstract

This draft specifies a Border Gateway Protocol (BGP) Network Layer Reachability Information (NLRI) encoding format for flow specifications (RFC 5575bis) that can match on a variety of tunneled traffic. In addition, flow specification components are specified for certain tunneling header fields.

Status of This Document

Table of Contents

1. Introduction

BGP Flow Specification (flowspec [RFC5575bis]) is an extension to BGP that supports the dissemination of traffic flow specification rules. It uses the BGP control plane to simplify the distribution of Access Control Lists (ACLs) and allows new filter rules to be injected to all BGP peers simultaneously without changing router configuration. A typical application of BGP flowspec is to automate the distribution of traffic filter lists to routers for Distributed Denial of Service (DDOS) mitigation.

BGP flowspec defines BGP Network Layer Reachability Information (NLRI) formats used to distribute traffic flow specification rules. AFI=1/SAFI=133 is for IPv4 unicast filtering. AFI=1/SAFI=134 is for IPv4 BGP/MPLS VPN filtering [RFC5575bis]. [FlowSpecV6] and [FlowSpecL2] extend the flowspec rules for IPv6 and Layer 2 Ethernet packets respectively. None of these previously defined flow specifications are suitable for matching in cases of tunneling or encapsulation where there might be duplicates of a layer of header such as two IPv6 headers in IP-in-IP [RFC2003] or a nested header sequence such as the Layer 2 and 3 headers encapsulated in VXLAN [RFC7348].

In the cloud computing era, multi-tenancy has become a core requirement for data centers. It is increasingly common to see tunneled traffic with a field to distinguish tenants. An example is the Network Virtualization Over Layer 3 (NVO3 [RFC8014]) overlay technology that can satisfy multi-tenancy key requirements. VXLAN [RFC7348] and NVGRE [RFC7637] are two typical NVO3 encapsulations. Other encapsulations such as IP-in-IP or GRE may be encountered. Because these tunnel / overlay technologies involving an additional level of encapsulation, flow specification that can match on the inner header as well as the outer header and fields in any tunneling header are needed.

In summary, Flow Specifications should be able to include inner nested header information as well as fields specific to the type of tunneling in use such as virtual network / tenant ID. This draft specifies methods for accomplishing this using SAFI=TBD1 and a new NLRI encoding.

1.1 Terminology

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.

The reader is assumed to be familiar with BGP terminology [RFC4271] [RFC4760]. The following terms and acronyms are used in this document with the meaning indicated:

ACL - Access Control List

DDoS - Distributed Denial of Service (Attack)

DSCP - Differentiated Services Code Point [RFC2474]

GRE - Generic Router Encapsulation [RFC2890]

L2TPv3 - Layer Two Tunneling Protocol - Version 3 [RFC3931]

NLRI - Network Layer Reachability Information [RFC4271] [RFC4760]

NVGRE - Network Virtualization Using Generic Routing Encapsulation [RFC7637]

NVO3 - Network Virtual Overlay Layer 3 [RFC8014]

PE - Provider Edge

VN - virtual network

VXLAN - Virtual eXtensible Local Area Network [RFC7348]

2. Tunneled Traffic Flow Specification NLRI

The Flowspec rules specified in [RFC5575bis], [FlowSpecV6], and [FlowSpecL2] cannot match or filter tunneled traffic based on the tunnel type, any tunnel header fields, or headers past the tunnel header. To enable flow specification of tunneled traffic, a new SAFI (TBD1) and NLRI encoding are specified. This encoding, shown in Figure 1, enables flow specification of more than one layer of header when needed.

```
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
         | Length                          2 octets      |
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
         | Tunnel Type                     2 octets      |
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
       Flags:
         +--+--+--+--+--+--+--+--+
         | D| I| Reserved        |         1 octet
         +--+--+--+--+--+--+--+--+
       Optional Routing Distinguisher:
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
         |                                               |
         +                                               +
         |                                               |
         + Routing Distinguisher           8 octets      +
         |                                               |
         +                                               +
         |                                               |
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
       Outer Flowspec:
         +--+--+--+--+--+--+--+--+
         | Outer Flowspec Length ...       1 or 2 octets
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
         | Outer Flowspec                  variable      :
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
       Tunnel Header Flowspec:
         +--+--+--+--+--+--+--+--+
         | Tunnel Flowspec Length ...      1 or 2 octets
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
         | Tunnel Header Flowspec          variable      :
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
       Optional Inner Flowspec:
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
         | Inner AFI                       2 octets      |
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
         | Inner Flowspec Length ...       1 or 2 octets
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
         | Inner Flowspec                  variable      :
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

Figure 1. Tunneled Traffic Flowspec NLRI

Length - The NLRI Length including the Tunnel Type encoded as an unsigned integer.

Tunnel Type - The type of tunnel using a value from the IANA BGP Tunnel Encapsulation Attribute Tunnel Types registry.

Flags: D bit - Indicates the presence of the Routing Distinguisher (see below).

Flags: I bit - Indicates the presence of the Inner AFI and the Inner Flowspec (see below).

Flags: Reserved - Six bits that MUST be sent as zero and ignored on receipt.

Routing Distinguisher - If the outer Layer 3 address belongs to a BGP/MPLS VPN, the routing distinguisher is included to indicate traffic filtering within that VPN. Because NVO3 outer layer addresses normally belong to a public network, a Route Distinguisher field is normally not needed for NVO3.

Outer Flowspec / Length - The flow specification for the outer header. The length is encoded as provided in Section 4.1 of [RFC5575bis]. The AFI for the Outer Flowspec is the AFI at the beginning of the BGP multiprotocol MP_REACH_NLRI or MP_UNREACH_NLRI containing the tunneled traffic flow specification NLRI.

Tunnel Header Flowspec / Length - The flow specification for the tunneling header. This specifies matching criterion on tunnel header fields as well as, implicitly, on the tunnel type which is indicated by the Tunnel Type field above. For some types of tunneling, such as IP-in-IP, there may be no tunnel header fields. For other types of tunneling, there may be several tunnel header fields on which matching can be specified with this flowspec.

Inner AFI - Depending on the Tunnel Type, there may be an Inner AFI that indicates the address family for the inner flow specification. There is no need for a SAFI as the treatment of this inner AFI and following flowspec is indicated by the outer SAFI TBD1, the SAFI for a tunneled traffic flow specification.

Inner Flowspec / Length - Depending on the Tunnel Type, there may be an inner flowspec for the header level encapsulated within the outer header. The length is encoded as provided in Section 4.1 of [RFC5575bis].

A Tunneled Traffic Flowspec matches if the Outer Flowspec, Tunnel Type, and Tunnel Header Flowspec match and, in addition, each of the following optional items that is present matches:

- Inner Flowspec, and
- Routing Distinguisher.

An omitted (as can be done for the Inner Flowspec) or null flowspec is considered to always match.

2.1 The SAFI Code Point

Use of the tunneled traffic flow specification NLRI format is indicated by SAFI=TBD1. This is used in conjunction with the AFI for the outer header, that is AFI=1 for IPv4, AFI=2 for IPv6, and AFI=6 for Layer 2.

2.2 Tunnel Header Component Code Points

For most cases of tunneled traffic, there are tunnel header fields that can be tested by components that appear in the Tunnel Header Flowspec field. The types for these components are specified in a Tunnel Header Flowspec component registry (see Section 6) and the initial entries in this registry are specified below.

All Tunnel Header field components defined below and all such components added in the future have a TLV structure as follows:

- one octet of type followed by
- one octet giving the length as an unsigned integer number of octets followed by
- the specific matching operations/values as determined by the type.

Type 1 - VN ID
Encoding: <type (1 octet), length (1 octet), [op, value]+>.

Defines a list of {operation, value} pairs used to match the 24-bit VN ID that is used as the tenant identification in some tunneling headers. For VXLAN encapsulation, the VN ID is the VNI. For NVGRE encapsulation, the VN ID is the VSID. op is encoded as specified in Section 4.2.3 of [RFC5575bis]. Values are encoded as a 1, 2, or 4 octet quantity. If value is 24-bits, it is left-justified in the first 3 octets of the value and the last value octet MUST be sent as zero and ignored on receipt.

Type 2 - Flow ID
Encoding: <type (1 octet), length (1 octet), [op, value]+>

Defines a list of {operation, value} pairs used to match 8-bit Flow ID fields which are currently only useful for NVGRE encapsulation. op is encoded as specified in Section 4.2.3 of [RFC5575bis]. Values are encoded as a 1-octet quantity.

Type 3 - Session
Encoding: <type (1 octet), length (1 octet), [op, value]+>

Defines a list of {operation, value} pairs used to match a

32-bit Session field. This field is called Key in GRE [RFC2890] encapsulation and Session ID in L2TPv3 encapsulation. op is encoded as specified in Section 4.2.3 of [RFC5575bis]. Values are encoded as a 1, 2, or 4 octet quantity.

Type 4 - Cookie
Encoding: <type (1 octet), length (1 octet), [op, value]+>

Defines a list of {operation, value} pairs used to match a variable length Cookie field. This is only useful in L2TPv3 encapsulation. op is encoded as specified in Section 4.2.3 of [RFC5575bis]. Values are encoded as a 1, 2, 4, or 8 octet quantity. If the Cookie does not fit exactly into the value length, it is left justified and padded with following octets that MUST be sent as zero and ignored on receipt.

Type 5 - Tunnel Header Flags
Encoding: <type (1 octet), length (1 octet), [op, bitmask]+>

Defines a list of {operation, value} pairs used to match against the tunnel header flags field. op is encoded as in Section 4.2.9 of [RFC5575bis]. bitmask is encoded as 1 octet for VXLAN-GPE and 2 octets for L2TP control messages. When matching on L2TP control message flags, the 3-bit Version subfield is treated as if it was zero.

Type 6 - L2TP Control Version
Encoding: <type (1 octet), length (1 octet), [op, value]+>

Defines a list of {operation, value} pairs used to match against the L2TPv3 Control Message Version. op is encoded as in Section 4.2.3 of [RFC5575bis]. Value is encoded as 1 octet.

Type 7 - L2TPv3 Control Connection ID
Encoding: <type (1 octet), length (1 octet), [op, value]+>

Defines a list of {operation, value} pairs used to match against the L2TPv3 Control Connection ID. op is encoded as in Section 4.2.3 of [RFC5575bis]. Value is encoded as 4 octets.

Type 8 - L2TPv3 Ns
Encoding: <type (1 octet), length (1 octet), [op, value]+>

Defines a list of {operation, value} pairs used to match against the L2TPv3 control message Ns field. op is encoded as in Section 4.2.3 of [RFC5575bis]. Value is encoded as 2 octets.

Type 9 - L2TPv3 Nr
Encoding: <type (1 octet), length (1 octet), [op, value]+>

Defines a list of {operation, value} pairs used to match against the L2TPv3 control message Nr field. op is encoded as in Section 4.2.3 of [RFC5575bis]. Value is encoded as 2 octets.

2.3 Specific Tunnel Types

The following subsections describe how to handle flow specification for several specific tunnel types.

2.3.1 VXLAN

The headers on a VXLAN [RFC7348] data packet are an outer Ethernet header, an outer IP header, a UDP header, the VXLAN header, and an inner Ethernet header. This inner Ethernet header is frequently, but not always, followed by an inner IP header. If the tunnel type is VXLAN, the I flag MUST be set in the Tunneled Traffic Flow Specification.

If the outer Ethernet header is not being matched, the version (IPv4 or IPv6) of the outer IP header is indicated by the AFI at the beginning of the multiprotocol MP_REACH_NLRI or MP_UNREACH_NLRI containing the Tunneled Traffic Flow Specification NLRI. The outer Flowspec is used to filter the outer headers including, if desired, the UDP header.

If the outer Ethernet header is being matched, then the initial AFI is 6 [FlowSpecL2] and the Outer Flowspec can match the outer Ethernet header, specify the IP version of the outer IP header, and match that IP header including, if desired, the UDP header.

The Tunnel Header Flowspec can be used to filter on the VXLAN header VN ID (VNI).

The Inner Flowspec can be used on the Inner Ethernet header [FlowSpecL2] and any following IP header. If the inner AFI is 6, then the inner Flowspec provides filtering of the Layer 2 header, indicates whether filtering on a following IPv4 or IPv6 header is desired, and if it is desired provides the Flowspec components for that filtering. If the Inner AFI is 1 or 2, the Inner Ethernet header is not matched and to match the Flowspec the Inner Ethernet header must be followed by an IPv4 or IPv6 header, respectively, and the inner Flowspec is used to filter that inner IP header.

The inner MAC/IP address is associated with the VN ID. In the NVO3 terminating into a VPN scenario, if multiple access VN IDs map to one VPN instance, one shared VN ID can be carried in the flowspec rule to enforce the rule on the entire VPN instance and the shared VN ID and VPN correspondence should be configured on each VPN PE beforehand. In this case, the function of the Layer 3 VN ID is the same as a Route Distinguisher: it acts as the identification of the VPN instance.

2.3.2 VXLAN-GPE

VXLAN-GPE [GPE] is similar to VXLAN. The VXLAN-GPE header is the same size as the VXLAN header but has been extended from the VXLAN header by specifying a number of bits that are reserved in the VXLAN header. In particular, a number of additional flag bits are specified and a Next Protocol field is added that is valid if the P flag bit is set in the VXLAN-GPE header. These flags bits can be tested using the Tunnel Header Flags flowspec component defined above. VXLAN and VXLAN-GPE are distinguished by the port number in the UDP header the precedes the VXLAN or VXLAN-GPE headers.

If the VXLAN-GPE header P flag is zero, then that header is followed by the same sequence as for VXLAN and the same flowspec choices apply (see Section 2.3.1).

If the VXLAN-GPE header P flag is one and that header’s next protocol field is 1, then the VXLAN-GPE header is followed by an IPv4 header (there is no Inner Ethernet header). The Inner Flowspec matches only if the Inner AFI is 1 and the Inner Flowspec matches.

If the VXLAN-GPE header P flag is one and that header’s next protocol field is 2, then the VXLAN-GPE header is followed by an IPv6 header (there is no Inner Ethernet header). The Inner Flowspec match only if the Inner AFI is 2 and the Inner Flowspec matches.

2.3.3 NVGRE

NVGRE [RFC7637] is similar to VXLAN except that the UDP header and VXLAN header immediately after the outer IP header are replaced by a GRE (Generic Router Encapsulation) header. The GRE header as used in NVGRE has no Checksum or Reserved1 field as shown in [RFC2890] but there are Virtual Subnet ID and Flow ID fields in place of what is labeled in [RFC2890] as the Key field. Processing and restrictions for NVGRE are as in Section 2.3.1 eliminating references to a UDP header and replacing references to the VXLAN header and its VN ID with references to the GRE header and its VN ID (VSID) and Flow ID.

2.3.4 L2TPv3

The headers on an L2TPv3 [RFC3931] packets are an outer Ethernet header, an outer IP header, the L2TPv3 header, an inner Ethernet header, and possibly an inner IP header if indicated by the inner Ethernet header EtherType. The Outer Flowspec operates on the outer headers that precede the L2TP Session Header. The version of IP in the outer IP header is specified by either the outer AFI at the beginning of the MP_REACH_NLRI or MP_UNREACH_NLRI or, if that AFI is 6 (L2), optionally specified by the inner AFI within that L2 flowspec.

L2TPv3 data messages and control messages both start with a Session ID and are distinguished by whether the Session ID is non-zero or zero, respectively. Data message filtering is further specified in Section 2.3.4.1 and control message filtering is further specified in Section 2.3.4.2.

2.3.4.1 L2TPv3 Data Messages

For data messages, the L2TPv3 Session Header consists of a 32-bit non-zero Session ID followed by a variable length Cookie (maximum length 8 octets). A Tunnel Header flowspec is assumed to apply to data messages unless the first component requires a zero Session ID.

The Session ID and Cookie can be filtered on by using the Session and Cookie flowspec components in the Tunnel Header Flowspec. To filter on Cookie or even be able to bypass Cookie and parse the remainder of the L2TPv3 packet, the node implementing flowspec needs to know the length and/or value of the Cookie fields of interest. This is negotiated at L2TPv3 session establishment and it is out of scope for this document how the node would learn this information. Of course, if flowspec is being used for DDOS mitigation and the Cookie has a fixed length and/or value in the DDOS traffic, this could be learned by inspecting that traffic.

If the I flag bit is zero, then no filtering is done on data beyond the L2TPv3 header. If the I flag is one, indicating the presence of an Inner Flowspec, and the node implementing flowspec does not know the length of the L2TPv3 header Cookie, the match fails. If that node does know the length of that Cookie, the Inner Flowspec if matched against the headers at the beginning of that data using the Inner AFI. If that Inner AFI is 1 or 2, then an inner IP header is required and filtering can be done on that IPv4 or IPv6 header respectively. If the Inner AFI is 6, filtering is done on the inner Ethernet header and, if an IPv4 or IPv6 inner AFI is specified within the inner L2 flowspec, done on the following IP header [FlowSpecL2].

2.3.4.2 L2TPv3 Control Messages

Control messages are distinguished by starting with a zero 32-bit Session ID. L2TPv3 control message flowspecs MUST start with a Session component that requires Session to be zero. For L2TPv3 control messages, there is no Cookie but there are L2TPv3 flags, a 3-bit Version field, a 32-bit Control Connection ID, and 16-bit Ns and Nr sequence numbers. These can be tested using the Tunnel Header Flags, L2TP Control Version, L2TPv3 Control Connection ID, L2TPv3 Ns, and L2TPv3 Nr flowspec components for the Tunnel Header Flowspec.

2.3.5 GRE

Generic Router Encapsulation (GRE [RFC2890]) is another type of encapsulation. The Outer Flowspec operates on the outer headers that precede the GRE header. The version of IP is specified by the outer AFI at the beginning of the MP_REACH_NLRI or MP_UNREACH_NLRI.

If the I flag bit is zero, no filtering is done on data after the GRE header. If the I flag bit is one in the tunnel flowspec, then there is an inner AFI and inner flowspec and the Protocol Type field of the GRE header must match the Inner AFI as follows for the tunnel Flowspec to match:

GRE Protocol Type	Inner AFI
0x0800 (IPv4)	1
0x86DD (IPv6)	2
0x6558	6

With the I flag a one and the Inner AFI and GRE Protocol Type fields match, the Inner Flowspec is used to filter the inner IP headers (Inner AFI=1 or 2) or the inner Ethernet header and optionally a following IP header (Inner AFI=6).

2.3.6 IP-in-IP

IP-in-IP encapsulation [RFC2003] is indicated when an outer IP header indicates an inner IP IPv4 or IPv6 header by the value of the outer IP header’s Protocol (IPv4) or Next Protocol (IPv6) field.

The IP version of the outer IP header (IPv4 or IPv6) matched is indicated by an AFI of 1 or 2 at the beginning of the MP_REACH_NLRI or MP_UNREACH_NLRI while if that AFI is 6, it indicates a match on the out Ethernet header and, optionally, the following IP Header [FlowSpecL2]. The IP version of the inner IP header is indicated by the Inner AFI and the Inner Flowspec applies to the inner IP header.

There are no fields that can be matched by the Tunnel Header Flowspec in the case of IP-in-IP.

2.4 Tunneled Traffic Actions

The traffic filtering actions previously specified in [RFC5575bis] and [FlowSpecL2] are used for tunneled traffic. For Traffic Marking in NVO3, only the DSCP in the outer header can be modified.

3. Order of Traffic Filtering Rules

The following rules determine which flowspec takes precedence where one or more are applicable and at least one of the applicable flowspecs is a tunneled traffic flowspec:

- In comparing an applicable tunneled traffic flow specification with an applicable non-tunneled flow specification, the tunneled specification has precedence.

- If comparing tunneled traffic flow specifications, if all are applicable, the tunnel types will be the same. Any that have a Routing Distinguisher will take precedence over those without a Routing Distinguisher. Of those with a Routing Distinguisher, all applicable flowspecs will have the same Routing Distinguisher.

- At this point in the process, all remaining contenders for the highest precedence will either not have a Routing Distinguisher or have equal Routing Distinguishers. If more than one contender remain, those with an L2 Outer Flowspec take precedence over those with an L3 Outer Flowspec. If the Outer Flowspec AFI is the same, their order of precedence is determined by comparing the Outer Flowspecs as described in [RFC5575bis] and [FlowSpecV6] for AFI for 1 or 2 respectively or [FlowSpecL2] for AFI=6.

- If the Outer Flowspecs are equal, then the Tunnel Header Flowspecs are compared using the usual sequential component comparison process [RFC5575bis].

- If the Tunnel Header Flowspecs are equal then compare the "I" flag. Those with an Inner Flowspec take precedence over those without an Inner Flowspec. If you get to this stage in the ordering process, those without an Inner Flowspec are equal. For those with an Inner Flowspec, check the Inner AFI. An L2 Inner AFI (AFI=6) takes precedence over an L3 Inner AFI.

- If the Inner AFIs are equal, precedence is determined by comparing the Inner Flowspecs as described in [FlowSpecL2] for L2 or [RFC5575bis] for L3.

4. Flow Spec Validation

Flowspecs received over AFI=1/SAFI=TBD1 or AFI=2/SFAI=TBD1 are validated, using only the Outer Flowspec, against routing reachability received over AFI=1/SAFI=133 and AFI=2/SAFI=133 respectively, as modified by [FlowSpecOID].

5. Security Considerations

No new security issues are introduced to the BGP protocol by this specification.

For general Flowspec security considerations, see [rfc5575bis].

6. IANA Considerations

IANA is requested to assign a new SAFI as follows:

Value	Description	Reference
TBD1	Tunneled traffic flow specification rules	[This document]

IANA is requested to create a Tunnel Header Flow Spec Component Type registry on the Flow Spec Component Types registries web page as follows:

Name: Tunnel Flow Spec Component Types
Reference: [this document]
Registration Procedures:

0	Reserved
1-127	Specification Required
128-254	First Come First Served
255	Reserved

Initial contents:

Type	Name	Reference
0	reserved	[this document]
1	VN ID	[this document]
2	Flow ID	[this document]
3	Session	[this document]
4	Cookie	[this document]
5	Tunnel Header Flags	[this document]
6	L2TP Control Version	[this document]
7	L2TPv3 Control Connection ID	[this document]
8	L2TPv3 Ns	[this document]
9	L2TPv3 Nr	[this document]
10-254	unassigned	[this document]
255	reserved	[this document]

Normative References

[RFC2003] - Perkins, C., "IP Encapsulation within IP", RFC 2003, DOI 10.17487/RFC2003, October 1996, <https://www.rfc-editor.org/info/rfc2003>.

[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>.

[RFC2474] - Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/info/rfc2474>.

[RFC2890] - Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, DOI 10.17487/RFC2890, September 2000, <https://www.rfc-editor.org/info/rfc2890>.

[RFC3931] - Lau, J., Ed., Townsley, M., Ed., and I. Goyret, Ed., "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, DOI 10.17487/RFC3931, March 2005, <https://www.rfc-editor.org/info/rfc3931>.

[RFC4271] - Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[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>.

[RFC7348] - Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[RFC7637] - Garg, P., Ed., and Y. Wang, Ed., "NVGRE: Network Virtualization Using Generic Routing Encapsulation", RFC 7637, DOI 10.17487/RFC7637, September 2015, <https://www.rfc-editor.org/info/rfc7637>.

[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>.

[FlowSpecL2] - W. Hao, et al, "Dissemination of Flow Specification Rules for L2 VPN", draft-ietf-idr-flowspec-l2vpn, work in progress.

[FlowSpecOID] - J. Uttaro, J. Alcaide, C. Filsfils, D. Smith, P. Mohapatra, "Revised Validation Procedure for BGP Flow Specifications", draft-ietf-idr-bgp-flowspec-oid, work in progress.

[FlowSpecV6] - R. Raszuk, et al, "Dissemination of Flow Specification Rules for IPv6", draft-ietf-idr-flow-spec-v6, work in progress.

[RFC5575bis] - Hares, S., Loibl, C., Raszuk, R., McPherson, D., Bacher, M., "Dissemination of Flow Specification Rules", draft-ietf-idr-rfc5575bis, work in progress.

Informative References

[RFC8014] - Black, D., Hudson, J., Kreeger, L., Lasserre, M., and T. Narten, "An Architecture for Data-Center Network Virtualization over Layer 3 (NVO3)", RFC 8014, DOI 10.17487/RFC8014, December 2016, <https://www.rfc-editor.org/info/rfc8014>.

[GPE] - P. Quinn, et al, "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe, work in progress.

Acknowledgments

The authors wish to acknowledge the important contributions of Jeff Haas, Susan Hares, Qiandeng Liang, Nan Wu, Yizhou Li, Robert Raszuk, and Lucy Yong.

Authors' Addresses

Donald Eastlake
Futurewei Technologies
2386 Panoramic Circle
Apopka, FL 32703 USA

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

Weiguo Hao
Huawei Technologies
101 Software Avenue,
Nanjing 210012 China

Email: haoweiguo@huawei.com

Shunwan Zhuang
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095 China

Email: zhuangshunwan@huawei.com

Zhenbin Li
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095 China

Email: lizhenbin@huawei.com

Rong Gu
China Mobile

Email: gurong_cmcc@outlook.com

Copyright, Disclaimer, and Additional IPR Provisions

Inter-Domain Routing
Internet-Draft
Obsoletes: 7752 (if approved)
Intended status: Standards Track
Expires: May 6, 2021

K. Talaulikar, Ed.
Cisco Systems

November 2, 2020

Distribution of Link-State and Traffic Engineering Information Using BGP
draft-ietf-idr-rfc7752bis-05

Abstract

In a number of environments, a component external to a network is called upon to perform computations based on the network topology and current state of the connections within the network, including Traffic Engineering (TE) information. This is information typically distributed by IGP routing protocols within the network.

This document describes a mechanism by which link-state and TE information can be collected from networks and shared with external components using the BGP routing protocol. This is achieved using a new BGP Network Layer Reachability Information (NLRI) encoding format. The mechanism is applicable to physical and virtual IGP links. The mechanism described is subject to policy control.

Applications of this technique include Application-Layer Traffic Optimization (ALTO) servers and Path Computation Elements (PCEs).

This document obsoletes RFC 7752 by completely replacing that document. It makes a number of small changes and clarifications to the previous specification.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The contents of a Link-State Database (LSDB) or of an IGP's Traffic Engineering Database (TED) describe only the links and nodes within an IGP area. Some applications, such as end-to-end Traffic Engineering (TE), would benefit from visibility outside one area or Autonomous System (AS) in order to make better decisions.

The IETF has defined the Path Computation Element (PCE) [RFC4655] as a mechanism for achieving the computation of end-to-end TE paths that cross the visibility of more than one TED or that require CPU-intensive or coordinated computations. The IETF has also defined the ALTO server [RFC5693] as an entity that generates an abstracted network topology and provides it to network-aware applications.

Both a PCE and an ALTO server need to gather information about the topologies and capabilities of the network in order to be able to fulfill their function.

This document describes a mechanism by which link-state and TE information can be collected from networks and shared with external components using the BGP routing protocol [RFC4271]. This is achieved using a new BGP Network Layer Reachability Information (NLRI) encoding format. The mechanism is applicable to physical and virtual links. The mechanism described is subject to policy control.

A router maintains one or more databases for storing link-state information about nodes and links in any given area. Link attributes stored in these databases include: local/remote IP addresses, local/ remote interface identifiers, link metric and TE metric, link bandwidth, reservable bandwidth, per Class-of-Service (CoS) class reservation state, preemption, and Shared Risk Link Groups (SRLGs). The router's BGP process can retrieve topology from these LSDBs and distribute it to a consumer, either directly or via a peer BGP speaker (typically a dedicated Route Reflector), using the encoding specified in this document.

An illustration of the collection of link-state and TE information and its distribution to consumers is shown in the Figure 1 below.

```
               +-----------+
               | Consumer  |
               +-----------+
                     ^
                     |
               +-----------+             +-----------+
               |    BGP    |             |    BGP    |
               |  Speaker  |<----------->|  Speaker  |  +-----------+
               |    RR1    |             |    RRm    |  | Consumer  |
               +-----------+             +-----------+  +-----------+
                   ^   ^                       ^              ^
                   |   |                       |              |
             +-----+   +---------+             +---------+    |
             |                   |                       |    |
       +-----------+       +-----------+             +-----------+
       |    BGP    |       |    BGP    |             |    BGP    |
       |  Speaker  |       |  Speaker  |    . . .    |  Speaker  |
       |    R1     |       |     R2    |             |    Rn     |
       +-----------+       +-----------+             +-----------+
             ^                   ^                         ^
             |                   |                         |
            IGP                 IGP                       IGP
```

Figure 1: Collection of Link-State and TE Information

A BGP speaker may apply configurable policy to the information that it distributes. Thus, it may distribute the real physical topology from the LSDB or the TED. Alternatively, it may create an abstracted topology, where virtual, aggregated nodes are connected by virtual paths. Aggregated nodes can be created, for example, out of multiple routers in a Point of Presence (POP). Abstracted topology can also be a mix of physical and virtual nodes and physical and virtual links. Furthermore, the BGP speaker can apply policy to determine when information is updated to the consumer so that there is a reduction of information flow from the network to the consumers. Mechanisms through which topologies can be aggregated or virtualized are outside the scope of this document.

This document obsoletes [RFC7752] by completely replacing that document. It makes a number of small changes and clarifications to the previous specification as documented in Appendix A.

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. Motivation and Applicability

This section describes use cases from which the requirements can be derived.

2.1. MPLS-TE with PCE

As described in [RFC4655], a PCE can be used to compute MPLS-TE paths within a "domain" (such as an IGP area) or across multiple domains (such as a multi-area AS or multiple ASes).

- Within a single area, the PCE offers enhanced computational power that may not be available on individual routers, sophisticated policy control and algorithms, and coordination of computation across the whole area.

- If a router wants to compute a MPLS-TE path across IGP areas, then its own TED lacks visibility of the complete topology. That means that the router cannot determine the end-to-end path and cannot even select the right exit router (Area Border Router (ABR)) for an optimal path. This is an issue for large-scale networks that need to segment their core networks into distinct areas but still want to take advantage of MPLS-TE.

Previous solutions used per-domain path computation [RFC5152]. The source router could only compute the path for the first area because the router only has full topological visibility for the first area along the path, but not for subsequent areas. Per-domain path computation uses a technique called "loose-hop-expansion" [RFC3209] and selects the exit ABR and other ABRs or AS Border Routers (ASBRs) using the IGP-computed shortest path topology for the remainder of the path. This may lead to sub-optimal paths, makes alternate/back-up path computation hard, and might result in no TE path being found when one really does exist.

The PCE presents a computation server that may have visibility into more than one IGP area or AS, or may cooperate with other PCEs to perform distributed path computation. The PCE obviously needs access to the TED for the area(s) it serves, but [RFC4655] does not describe how this is achieved. Many implementations make the PCE a passive participant in the IGP so that it can learn the latest state of the network, but this may be sub-optimal when the network is subject to a high degree of churn or when the PCE is responsible for multiple areas.

The following figure shows how a PCE can get its TED information using the mechanism described in this document.

```
                +----------+                           +---------+
                |  -----   |                           |   BGP   |
                | | TED |<-+-------------------------->| Speaker |
                |  -----   |   TED synchronization     |         |
                |    |     |        mechanism:         +---------+
                |    |     | BGP with Link-State NLRI
                |    v     |
                |  -----   |
                | | PCE |  |
                |  -----   |
                +----------+
                     ^
                     | Request/
                     | Response
                     v
       Service  +----------+   Signaling  +----------+
       Request  | Head-End |   Protocol   | Adjacent |
       -------->|  Node    |<------------>|   Node   |
                +----------+              +----------+
```

Figure 2: External PCE Node Using a TED Synchronization Mechanism

The mechanism in this document allows the necessary TED information to be collected from the IGP within the network, filtered according to configurable policy, and distributed to the PCE as necessary.

2.2. ALTO Server Network API

An ALTO server [RFC5693] is an entity that generates an abstracted network topology and provides it to network-aware applications over a web-service-based API. Example applications are peer-to-peer (P2P) clients or trackers, or Content Distribution Networks (CDNs). The abstracted network topology comes in the form of two maps: a Network Map that specifies allocation of prefixes to Partition Identifiers (PIDs), and a Cost Map that specifies the cost between PIDs listed in the Network Map. For more details, see [RFC7285].

ALTO abstract network topologies can be auto-generated from the physical topology of the underlying network. The generation would typically be based on policies and rules set by the operator. Both prefix and TE data are required: prefix data is required to generate ALTO Network Maps, and TE (topology) data is required to generate ALTO Cost Maps. Prefix data is carried and originated in BGP, and TE data is originated and carried in an IGP. The mechanism defined in this document provides a single interface through which an ALTO server can retrieve all the necessary prefix and network topology data from the underlying network. Note that an ALTO server can use other mechanisms to get network data, for example, peering with multiple IGP and BGP speakers.

The following figure shows how an ALTO server can get network topology information from the underlying network using the mechanism described in this document.

```
     +--------+
     | Client |<--+
     +--------+   |
                  |    ALTO    +--------+     BGP with    +---------+
     +--------+   |  Protocol  |  ALTO  | Link-State NLRI |   BGP   |
     | Client |<--+------------| Server |<----------------| Speaker |
     +--------+   |            |        |                 |         |
                  |            +--------+                 +---------+
     +--------+   |
     | Client |<--+
     +--------+
```

Figure 3: ALTO Server Using Network Topology Information

3. BGP Speaker Roles for BGP-LS

In the illustration shown in Figure 1, the BGP Speakers can be seen playing different roles in the distribution of information using BGP-LS. This section introduces terms that explain the different roles of the BGP Speakers which are then used through the rest of this document.

- BGP-LS Producer: The BGP Speakers R1, R2, ... Rn, originate link-state information from their underlying link-state IGP protocols into BGP-LS. If R1 and R2 are in the same IGP area, then likely they are originating the same link-state information into BGP-LS. R1 may also source information from sources other than IGP, e.g. its local node information. The term BGP-LS Producer refers to the BGP Speaker that is originating link-state information into BGP.

- BGP-LS Consumer: The BGP Speakers RR1 and Rn are handing off the BGP-LS information that they have collected to a consumer application. The BGP protocol implementation and the consumer application may be on the same or different nodes. The term BGP-LS Consumer refers to the consumer application/process and not the BGP Speaker. This document only covers the BGP implementation. The consumer application and the design of interface between BGP and consumer application may be implementation specific and outside the scope of this document.

- BGP-LS Propagator: The BGP Speaker RRm propagates the BGP-LS information between the BGP Speaker Rn and the BGP Speaker RR1. The BGP implementation on RRm is doing the propagation of BGP-LS updates and performing BGP best path calculations. Similarly, the BGP Speaker RR1 is receiving BGP-LS information from R1, R2 and RRm and propagating the information to the BGP-LS Consumer after performing BGP best path calculations. The term BGP-LS Propagator refers to the BGP Speaker that is performing BGP protocol processing on the link-state information.

The above roles are not mutually exclusive. The same BGP Speaker may be the producer for some link-state information and propagator for some other link-state information while also providing this information to a consumer application. Nothing precludes a BGP implementation performing some of the validation and processing on behalf of the BGP-LS Consumer as long as it does not impact the semantics of its role as BGP-LS Propagator as described in this document.

The rest of this document refers to the role when describing procedures that are specific to that role. When the role is not specified, then the said procedure applies to all BGP Speakers.

4. Carrying Link-State Information in BGP

This specification contains two parts: definition of a new BGP NLRI that describes links, nodes, and prefixes comprising IGP link-state information and definition of a new BGP path attribute (BGP-LS Attribute) that carries link, node, and prefix properties and attributes, such as the link and prefix metric or auxiliary Router-IDs of nodes, etc.

It is desirable to keep the dependencies on the protocol source of this attribute to a minimum and represent any content in an IGP-neutral way, such that applications that want to learn about a link-state topology do not need to know about any OSPF or IS-IS protocol specifics.

This section mainly describes the procedures at a BGP-LS Producer that originate link-state information into BGP-LS.

4.1. TLV Format

Information in the new Link-State NLRIs and the BGP-LS Attribute is encoded in Type/Length/Value triplets. The TLV format is shown in Figure 4 and applies to both the NLRI and the BGP-LS Attribute encodings.

```
     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            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //                       Value (variable)                      //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: TLV Format

The Length field defines the length of the value portion in octets (thus, a TLV with no value portion would have a length of zero). The TLV is not padded to 4-octet alignment. Unknown and unsupported types MUST be preserved and propagated within both the NLRI and the BGP-LS Attribute. The presence of unrecognized or unexpected TLVs MUST NOT result in the NLRI or the BGP-LS Attribute being considered as malformed.

In order to compare NLRIs with unknown TLVs, all TLVs within the NLRI MUST be ordered in ascending order by TLV Type. If there are multiple TLVs of the same type within a single NLRI, then the TLVs sharing the same type MUST be in ascending order based on the value field. Comparison of the value fields is performed by treating the entire field as an opaque hexadecimal string. Standard string comparison rules apply. NLRIs having TLVs which do not follow the above ordering rules MUST be considered as malformed by a BGP-LS Propagator. This ensures that multiple copies of the same NLRI from multiple BGP-LS Producers and the ambiguity arising there from is prevented.

All TLVs within the NLRI that are not specified as mandatory are considered optional. All TLVs within the BGP-LS Attribute are considered optional unless specified otherwise.

The TLVs within the BGP-LS Attribute MAY be ordered in ascending order by TLV type. BGP-LS Attribute with unordered TLVs MUST NOT be considered malformed.

4.2. The Link-State NLRI

The MP_REACH_NLRI and MP_UNREACH_NLRI attributes are BGP's containers for carrying opaque information. This specification defines three Link-State NLRI types that describes either a node, a link, and a prefix.

All non-VPN link, node, and prefix information SHALL be encoded using AFI 16388 / SAFI 71. VPN link, node, and prefix information SHALL be encoded using AFI 16388 / SAFI 72.

In order for two BGP speakers to exchange Link-State NLRI, they MUST use BGP Capabilities Advertisement to ensure that they are both capable of properly processing such NLRI. This is done as specified in [RFC4760], by using capability code 1 (multi-protocol BGP), with AFI 16388 / SAFI 71 for BGP-LS, and AFI 16388 / SAFI 72 for BGP-LS-VPN.

New Link-State NLRI Types may be introduced in the future. Since supported NLRI type values within the address family are not expressed in the Multiprotocol BGP (MP-BGP) capability [RFC4760], it is possible that a BGP speaker has advertised support for Link-State but does not support a particular Link-State NLRI type. In order to allow introduction of new Link-State NLRI types seamlessly in the future, without the need for upgrading all BGP speakers in the propagation path (e.g. a route reflector), this document deviates from the default handling behavior specified by [RFC7606] for Link-State address-family. An implementation MUST handle unrecognized Link-State NLRI types as opaque objects and MUST preserve and propagate them.

The format of the Link-State NLRI is shown in the following figures.

```
     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)                 //
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Link-State AFI 16388 / SAFI 71 NLRI 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
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            NLRI Type          |     Total NLRI Length         |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     +                       Route Distinguisher                     +
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     //                  Link-State NLRI (variable)                 //
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Link-State VPN AFI 16388 / SAFI 72 NLRI Format

The Total NLRI Length field contains the cumulative length, in octets, of the rest of the NLRI, not including the NLRI Type field or itself. For VPN applications, it also includes the length of the Route Distinguisher.

Type	NLRI Type
1	Node NLRI
2	Link NLRI
3	IPv4 Topology Prefix NLRI
4	IPv6 Topology Prefix NLRI

Table 1: NLRI Types

Route Distinguishers are defined and discussed in [RFC4364].

The Node NLRI (NLRI Type = 1) is 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)                          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     //                Local Node Descriptors (variable)            //
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: The Node NLRI Format

The Link NLRI (NLRI Type = 2) is 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)                          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     //               Local Node Descriptors (variable)             //
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     //               Remote Node Descriptors (variable)            //
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     //                  Link Descriptors (variable)                //
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: The Link NLRI Format

The IPv4 and IPv6 Prefix NLRIs (NLRI Type = 3 and Type = 4) use the same 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)                          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     //              Local Node Descriptors (variable)              //
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     //                Prefix Descriptors (variable)                //
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: The IPv4/IPv6 Topology Prefix NLRI Format

The Protocol-ID field can contain one of the following values:

Protocol-ID	NLRI information source protocol
1	IS-IS Level 1
2	IS-IS Level 2
3	OSPFv2
4	Direct
5	Static configuration
6	OSPFv3

Table 2: Protocol Identifiers

The 'Direct' and 'Static configuration' protocol types SHOULD be used when BGP-LS is sourcing local information. For all information derived from other protocols, the corresponding Protocol-ID MUST be used. If BGP-LS has direct access to interface information and wants to advertise a local link, then the Protocol-ID 'Direct' SHOULD be used. For modeling virtual links, such as described in Section 5, the Protocol-ID 'Static configuration' SHOULD be used.

A router MAY run multiple protocol instances of OSPF or ISIS where by it becomes a border router between multiple IGP domains. Both OSPF and IS-IS MAY also run multiple routing protocol instances over the same link. See [RFC8202] and [RFC6549]. These instances define independent IGP routing domains. The 64-bit Identifier field carries a BGP-LS Instance Identifier (Instance-ID) that is used to identify the IGP routing domain where the NLRI belongs. The NLRIs representing link-state objects (nodes, links, or prefixes) from the same IGP routing instance MUST have the same Identifier field value. NLRIs with different Identifier field values MUST be considered to be from different IGP routing instances. The Identifier field value 0 is RECOMMENDED to be used when there is only a single protocol instance in the network where BGP-LS is operational.

An implementation which supports multiple IGP instances MUST support the configuration of unique BGP-LS Instance-IDs at the routing protocol instance level. The network operator MUST assign consistent BGP-LS Instance-ID values on all BGP-LS Producers within a given IGP domain. Unique BGP-LS Instance-ID values MUST be assigned to routing protocol instances operating in different IGP domains. This allows the BGP-LS Consumer to build an accurate segregated multi-domain topology based on the Identifier field even when the topology is advertised via BGP-LS by multiple BGP-LS Producers in the network.

When the above described semantics and recommendations are not followed, a BGP-LS Consumer may see duplicate link-state objects for the same node, link or prefix when there are multiple BGP-LS

Producers deployed. This may also result in the BGP-LS Consumers getting an inaccurate network-wide topology.

When adding, removing or modifying a TLV/sub-TLV from a Link-State NLRI, the BGP-LS Producer MUST withdraw the old NLRI by including it in the MP_UNREACH_NLRI. Not doing so can result in duplicate and inconsistent link-state objects hanging around in the BGP-LS table.

Each Node Descriptor, Link Descriptor and Prefix Descriptor consists of one or more TLVs, as described in the following sections. These Descriptor TLVs are applicable for the Node, Link and Prefix NLRI Types for the protocols listed in Table 2. Documents extending BGP-LS specifications with new NLRI Types and/or protocols MUST specify the NLRI Descriptors for them.

4.2.1. Node Descriptors

Each link is anchored by a pair of Router-IDs that are used by the underlying IGP, namely, a 48-bit ISO System-ID for IS-IS and a 32-bit Router-ID for OSPFv2 and OSPFv3. An IGP may use one or more additional auxiliary Router-IDs, mainly for Traffic Engineering purposes. For example, IS-IS may have one or more IPv4 and IPv6 TE Router-IDs [RFC5305] [RFC6119]. These auxiliary Router-IDs MUST be included in the node attribute described in Section 4.3.1 and MAY be included in link attribute described in Section 4.3.2. The advertisement of the TE Router-IDs help a BGP-LS Consumer to correlate multiple link-state objects (e.g. in different IGP instances or areas/levels) to the same node in the network.

It is desirable that the Router-ID assignments inside the Node Descriptor are globally unique. However, there may be Router-ID spaces (e.g., ISO) where no global registry exists, or worse, Router-IDs have been allocated following the private-IP allocation described in RFC 1918 [RFC1918]. BGP-LS uses the Autonomous System (AS) Number to disambiguate the Router-IDs, as described in Section 4.2.1.1.

4.2.1.1. Globally Unique Node/Link/Prefix Identifiers

One problem that needs to be addressed is the ability to identify an IGP node globally (by "globally", we mean within the BGP-LS database collected by all BGP-LS speakers that talk to each other). This can be expressed through the following two requirements:

(A) The same node MUST NOT be represented by two keys (otherwise, one node will look like two nodes).

(B) Two different nodes MUST NOT be represented by the same key (otherwise, two nodes will look like one node).

We define an "IGP domain" to be the set of nodes (hence, by extension links and prefixes) within which each node has a unique IGP representation by using the combination of Area-ID, Router-ID, Protocol-ID, Multi-Topology ID, and Instance-ID. The problem is that BGP may receive node/link/prefix information from multiple independent "IGP domains", and we need to distinguish between them. Moreover, we can’t assume there is always one and only one IGP domain per AS. During IGP transitions, it may happen that two redundant IGPs are in place.

The mapping of the Instance-ID to the Identifier field as described earlier along with a set of sub-TLVs described in Section 4.2.1.4, allows specification of a flexible key for any given node/link information such that global uniqueness of the NLRI is ensured.

4.2.1.2. Local Node Descriptors

The Local Node Descriptors TLV contains Node Descriptors for the node anchoring the local end of the link. This is a mandatory TLV in all three types of NLRIs (node, link, and prefix). The Type is 256. The length of this TLV is variable. The value contains one or more Node Descriptor Sub-TLVs defined in Section 4.2.1.4.

```
     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            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    //              Node Descriptor Sub-TLVs (variable)            //
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: Local Node Descriptors TLV Format

4.2.1.3. Remote Node Descriptors

The Remote Node Descriptors TLV contains Node Descriptors for the node anchoring the remote end of the link. This is a mandatory TLV for Link NLRIs. The type is 257. The length of this TLV is variable. The value contains one or more Node Descriptor Sub-TLVs defined in Section 4.2.1.4.

```
     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            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    //              Node Descriptor Sub-TLVs (variable)            //
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: Remote Node Descriptors TLV Format

4.2.1.4. Node Descriptor Sub-TLVs

The Node Descriptor Sub-TLV type code points and lengths are listed in the following table:

Sub-TLV Code Point	Description	Length
512	Autonomous System	4
513	BGP-LS Identifier (deprecated)	4
514	OSPF Area-ID	4
515	IGP Router-ID	Variable

Table 3: Node Descriptor Sub-TLVs

The sub-TLV values in Node Descriptor TLVs are defined as follows:

Autonomous System: Opaque value (32-bit AS Number). This is an optional TLV. The value SHOULD be set to the AS Number associated with the BGP process originating the link-state information. An implementation MAY provide a configuration option on the BGP-LS Producer to use a value different.

BGP-LS Identifier: Opaque value (32-bit ID). This is an optional TLV. In conjunction with Autonomous System Number (ASN), uniquely identifies the BGP-LS domain. The combination of ASN and BGP-LS ID MUST be globally unique. All BGP-LS speakers within an IGP flooding-set (set of IGP nodes within which an LSP/LSA is flooded) MUST use the same ASN, BGP-LS ID tuple. If an IGP domain consists of multiple flooding-sets, then all BGP-LS speakers within the IGP domain SHOULD use the same ASN, BGP-LS ID tuple.

Area-ID: Used to identify the 32-bit area to which the information advertised in the NLRI belongs. This is a mandatory TLV when originating information from OSPF that is derived from area-scope

LSAs. The Area Identifier allows different NLRIs of the same router to be discriminated on a per area basis. It is not used for NLRIs when carrying information that is derived from AS-scope LSAs as it is not associated with a specific area.

IGP Router-ID: Opaque value. This is a mandatory TLV when originating information from IS-IS, OSPF, direct or static. For an IS-IS non-pseudonode, this contains a 6-octet ISO Node-ID (ISO system-ID). For an IS-IS pseudonode corresponding to a LAN, this contains the 6-octet ISO Node-ID of the Designated Intermediate System (DIS) followed by a 1-octet, nonzero PSN identifier (7 octets in total). For an OSPFv2 or OSPFv3 non-pseudonode, this contains the 4-octet Router-ID. For an OSPFv2 pseudonode representing a LAN, this contains the 4-octet Router-ID of the Designated Router (DR) followed by the 4-octet IPv4 address of the DR’s interface to the LAN (8 octets in total). Similarly, for an OSPFv3 pseudonode, this contains the 4-octet Router-ID of the DR followed by the 4-octet interface identifier of the DR’s interface to the LAN (8 octets in total). The TLV size in combination with the protocol identifier enables the decoder to determine the type of the node. For Direct or Static configuration, the value SHOULD be taken from an IPv4 or IPv6 address (e.g. loopback interface) configured on the node.

There can be at most one instance of each sub-TLV type present in any Node Descriptor. The sub-TLVs within a Node Descriptor MUST be arranged in ascending order by sub-TLV type. This needs to be done in order to compare NLRIs, even when an implementation encounters an unknown sub-TLV. Using stable sorting, an implementation can do binary comparison of NLRIs and hence allow incremental deployment of new key sub-TLVs.

The BGP-LS Identifier was introduced by [RFC7752] and it’s use is being deprecated by this document. Implementations MUST continue to support this sub-TLV for backward compatibility. The default value of 0 is RECOMMENDED to be use when a BGP-LS Producer includes this sub-TLV when originating information into BGP-LS. Implementations MAY provide an option to configure this value for backward compatibility reasons. The use of the Instance-ID in the Identifier field is the RECOMMENDED way of segregation of different IGP domains in BGP-LS.

4.2.2. Link Descriptors

The Link Descriptor field is a set of Type/Length/Value (TLV) triplets. The format of each TLV is shown in Section 4.1. The Link Descriptor TLVs uniquely identify a link among multiple parallel links between a pair of anchor routers. A link described by the Link

Descriptor TLVs actually is a "half-link", a unidirectional representation of a logical link. In order to fully describe a single logical link, two originating routers advertise a half-link each, i.e., two Link NLRIs are advertised for a given point-to-point link.

A BGP-LS Consumer should not consider a link between two nodes as being available unless it has received the two Link NLRIs corresponding to the half-link representation of that link from both the nodes. This check is similar to the 'two way connectivity check' that is performed by link-state IGPs and is also required to be done by BGP-LS Consumers of link-state topology.

A BGP-LS Producer MAY supress the advertisement of a Link NLRI, corresponding to a half link, from a link-state IGP unless it has verified that the link is being reported in the IS-IS LSP or OSPF Router LSA by both the nodes connected by that link. This 'two way connectivity check' is performed by link-state IGPs during their computation and may be leveraged before passing information for any half-link that is reported from these IGPs in to BGP-LS. This ensures that only those Link State IGP adjacencies which are established get reported via Link NLRIs. Such a 'two way connectivity check' may be also required in certain cases (e.g. with OSPF) to obtain the proper link identifiers of the remote node.

The format and semantics of the Value fields in most Link Descriptor TLVs correspond to the format and semantics of Value fields in IS-IS Extended IS Reachability sub-TLVs, defined in [RFC5305], [RFC5307], and [RFC6119]. Although the encodings for Link Descriptor TLVs were originally defined for IS-IS, the TLVs can carry data sourced by either IS-IS or OSPF.

The following TLVs are defined as Link Descriptors in the Link NLRI:

TLV Code Point	Description	IS-IS TLV/Sub-TLV	Reference (RFC/Section)
258	Link Local/Remote Identifiers	22/4	[RFC5307] / 1.1
259	IPv4 interface address	22/6	[RFC5305] / 3.2
260	IPv4 neighbor address	22/8	[RFC5305] / 3.3
261	IPv6 interface address	22/12	[RFC6119] / 4.2
262	IPv6 neighbor address	22/13	[RFC6119] / 4.3
263	Multi-Topology Identifier	---	Section 4.2.2.1

Table 4: Link Descriptor TLVs

The information about a link present in the LSA/LSP originated by the local node of the link determines the set of TLVs in the Link Descriptor of the link.

If interface and neighbor addresses, either IPv4 or IPv6, are present, then the IP address TLVs MUST be included and the Link Local/Remote Identifiers TLV MUST NOT be included in the Link Descriptor. The Link Local/Remote Identifiers TLV MAY be included in the link attribute when available. IPv6 link-local addresses MUST NOT be carried in the IPv6 address TLVs as descriptors of a link as they are not considered unique.

If interface and neighbor addresses are not present and the link local/remote identifiers are present, then the Link Local/Remote Identifiers TLV MUST be included in the Link Descriptor. The Link Local/Remote Identifiers MUST be included in the Link Descriptor also in the case of links having only IPv6 link-local addressing on them.

The Multi-Topology Identifier TLV MUST be included in Link Descriptor if the underlying IGP link object is associated with a non-default topology.

The TLVs/sub-TLVs corresponding to the interface addresses and/or the local/remote identfiers may not always be signaled in the IGPs unless their advertisement is enabled specifically. In such cases, a BGP-LS Producer may not be able to generate valid Link NLRIs for such link advertisements from the IGPs.

4.2.2.1. Multi-Topology ID

The Multi-Topology ID (MT-ID) TLV carries one or more IS-IS or OSPF Multi-Topology IDs for a link, node, or prefix.

Semantics of the IS-IS MT-ID are defined in Section 7.1 and 7.2 of RFC 5120 [RFC5120]. Semantics of the OSPF MT-ID are defined in Section 3.7 of RFC 4915 [RFC4915]. If the value in the MT-ID TLV is derived from OSPF, then the upper 5 bits of the MT-ID field MUST be set to 0.

The format of the MT-ID TLV is 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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Type             |          Length=2*n           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |R R R R|  Multi-Topology ID 1  |             ....             //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //             ....             |R R R R|  Multi-Topology ID n  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12: Multi-Topology ID TLV Format

where Type is 263, Length is 2*n, and n is the number of MT-IDs carried in the TLV.

The MT-ID TLV MAY be present in a Link Descriptor, a Prefix Descriptor, or the BGP-LS attribute of a Node NLRI. In a Link or Prefix Descriptor, only a single MT-ID TLV containing the MT-ID of the topology where the link or the prefix is reachable is allowed. In case one wants to advertise multiple topologies for a given Link Descriptor or Prefix Descriptor, multiple NLRIs MUST be generated where each NLRI contains a single unique MT-ID. When used in the Link or Prefix Descriptor TLV for IS-IS, the Bits R are reserved and MUST be set to 0 (as per Section 7.2 of RFC 5120 [RFC5120]) when originated and ignored on receipt.

In the BGP-LS attribute of a Node NLRI, one MT-ID TLV containing the array of MT-IDs of all topologies where the node is reachable is allowed. When used in the Node Attribute TLV for IS-IS, the Bits R are set as per Section 7.1 of RFC 5120 [RFC5120].

4.2.3. Prefix Descriptors

The Prefix Descriptor field is a set of Type/Length/Value (TLV) triplets. Prefix Descriptor TLVs uniquely identify an IPv4 or IPv6 prefix originated by a node. The following TLVs are defined as Prefix Descriptors in the IPv4/IPv6 Prefix NLRI:

TLV Code Point	Description	Length	Reference (RFC/Section)
263	Multi-Topology Identifier	variable	Section 4.2.2.1
264	OSPF Route Type	1	Section 4.2.3.1
265	IP Reachability Information	variable	Section 4.2.3.2

Table 5: Prefix Descriptor TLVs

The Multi-Topology Identifier TLV MUST be included in Prefix Descriptor if the underlying IGP prefix object is associated with a non-default topology.

4.2.3.1. OSPF Route Type

The OSPF Route Type TLV is a mandatory TLV corresponding to Prefix NLRIs originated from OSPF. It is used to identify the OSPF route type of the prefix. An OSPF prefix MAY be advertised in the OSPF domain with multiple route types. The Route Type TLV allows the discrimination of these advertisements. The format of the OSPF Route Type TLV is 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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Type             |             Length            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Route Type   |
    +-+-+-+-+-+-+-+-+
```

Figure 13: OSPF Route Type TLV Format

where the Type and Length fields of the TLV are defined in Table 5. The OSPF Route Type field values are defined in the OSPF protocol and can be one of the following:

- Intra-Area (0x1)

- Inter-Area (0x2)

- External 1 (0x3)

- External 2 (0x4)

- NSSA 1 (0x5)

- NSSA 2 (0x6)

4.2.3.2. IP Reachability Information

The IP Reachability Information TLV is a mandatory TLV for IPv4 & IPv6 Prefix NLRI types. The TLV contains one IP address prefix (IPv4 or IPv6) originally advertised in the IGP topology. Its purpose is to glue a particular BGP service NLRI by virtue of its BGP next hop to a given node in the LSDB. A router SHOULD advertise an IP Prefix NLRI for each of its BGP next hops. The format of the IP Reachability Information TLV is 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
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |              Type             |             Length            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | Prefix Length | IP Prefix (variable)                         //
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14: IP Reachability Information TLV Format

The Type and Length fields of the TLV are defined in Table 5. The following two fields determine the reachability information of the address family. The Prefix Length field contains the length of the prefix in bits. The IP Prefix field contains the most significant octets of the prefix, i.e., 1 octet for prefix length 1 up to 8, 2 octets for prefix length 9 to 16, 3 octets for prefix length 17 up to 24, 4 octets for prefix length 25 up to 32, etc.

4.3. The BGP-LS Attribute

The BGP-LS Attribute is an optional, non-transitive BGP attribute that is used to carry link, node, and prefix parameters and attributes. It is defined as a set of Type/Length/Value (TLV) triplets, described in the following section. This attribute SHOULD only be included with Link-State NLRIs. This attribute MUST be ignored for all other address families.

The Node Attribute TLVs, Link Attribute TLVs and Prefix Attribute TLVs are sets of TLVs that may be encoded in the BGP-LS Attribute associated with a Node NLRI, Link NLRI and Prefix NLRI respectively.

The BGP-LS Attribute may potentially grow large in size depending on the amount of link-state information associated with a single Link-State NLRI. The BGP specification [RFC4271] mandates a maximum BGP message size of 4096 octets. It is RECOMMENDED that an implementation support [RFC8654] in order to accommodate larger size of information within the BGP-LS Attribute. BGP-LS Producers MUST ensure that they limit the TLVs included in the BGP-LS Attribute to ensure that a BGP update message for a single Link-State NLRI does not cross the maximum limit for a BGP message. The determination of the types of TLVs to be included MAY be made by the BGP-LS Producer based on the BGP-LS Consumer applications requirement and is outside the scope of this document. When a BGP-LS Propagator finds that it is exceeding the maximum BGP message size due to addition or update of some other BGP Attribute (e.g. AS_PATH), it MUST consider the BGP-LS Attribute to be malformed and handle the propagation as described in Section 7.2.2.

4.3.1. Node Attribute TLVs

The following Node Attribute TLVs are defined for the BGP-LS Attribute associated with a Node NLRI:

TLV Code Point	Description	Length	Reference (RFC/Section)
263	Multi-Topology Identifier	variable	Section 4.2.2.1
1024	Node Flag Bits	1	Section 4.3.1.1
1025	Opaque Node Attribute	variable	Section 4.3.1.5
1026	Node Name	variable	Section 4.3.1.3
1027	IS-IS Area Identifier	variable	Section 4.3.1.2
1028	IPv4 Router-ID of Local Node	4	[RFC5305] / 4.3
1029	IPv6 Router-ID of Local Node	16	[RFC6119] / 4.1

Table 6: Node Attribute TLVs

4.3.1.1. Node Flag Bits TLV

The Node Flag Bits TLV carries a bit mask describing node attributes. The value is a 1 octet length bit array of flags, where each bit represents a node operational state or attribute.

```
     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            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |O|T|E|B|R|V|   |
    +-+-+-+-+-+-+-+-+
```

Figure 15: Node Flag Bits TLV Format

The bits are defined as follows:

Bit	Description	Reference
'O'	Overload Bit	[ISO10589]
'T'	Attached Bit	[ISO10589]
'E'	External Bit	[RFC2328]
'B'	ABR Bit	[RFC2328]
'R'	Router Bit	[RFC5340]
'V'	V6 Bit	[RFC5340]

Table 7: Node Flag Bits Definitions

4.3.1.2. IS-IS Area Identifier TLV

An IS-IS node can be part of one or more IS-IS areas. Each of these area addresses is carried in the IS-IS Area Identifier TLV. If multiple area addresses are present, multiple TLVs are used to encode them. The IS-IS Area Identifier TLV may be present in the BGP-LS attribute only when advertised in the Link-State Node NLRI.

```
     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            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //                Area Identifier (variable)                   //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 16: IS-IS Area Identifier TLV Format

4.3.1.3. Node Name TLV

The Node Name TLV is optional. Its structure and encoding has been borrowed from [RFC5301]. The Value field identifies the symbolic name of the router node. This symbolic name can be the Fully Qualified Domain Name (FQDN) for the router, it can be a subset of the FQDN (e.g., a hostname), or it can be any string operators want to use for the router. The use of FQDN or a subset of it is strongly RECOMMENDED. The maximum length of the Node Name TLV is 255 octets.

The Value field is encoded in 7-bit ASCII. If a user interface for configuring or displaying this field permits Unicode characters, that user interface is responsible for applying the ToASCII and/or ToUnicode algorithm as described in [RFC5890] to achieve the correct format for transmission or display.

[RFC5301] describes an IS-IS-specific extension and [RFC5642] describes an OSPF extension for advertisement of Node Name which MAY encoded in the Node Name TLV.

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

Figure 17: Node Name Format

4.3.1.4. Local IPv4/IPv6 Router-ID TLVs

The local IPv4/IPv6 Router-ID TLVs are used to describe auxiliary Router-IDs that the IGP might be using, e.g., for TE and migration purposes such as correlating a Node-ID between different protocols. If there is more than one auxiliary Router-ID of a given type, then each one is encoded in its own TLV.

4.3.1.5. Opaque Node Attribute TLV

The Opaque Node Attribute TLV is an envelope that transparently carries optional Node Attribute TLVs advertised by a router. An originating router shall use this TLV for encoding information specific to the protocol advertised in the NLRI header Protocol-ID field or new protocol extensions to the protocol as advertised in the NLRI header Protocol-ID field for which there is no protocol-neutral representation in the BGP Link-State NLRI. The primary use of the Opaque Node Attribute TLV is to bridge the document lag between, e.g., a new IGP link-state attribute being defined and the protocol-neutral BGP-LS extensions being published.

In the case of OSPF, this TLV MAY be used to advertise information carried using the TLVs in the "OSPF Router Information (RI) TLVs" registry [RFC7770] under the IANA OSPF Parameters registry.

```
     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            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //              Opaque node attributes (variable)              //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 18: Opaque Node Attribute Format

4.3.2. Link Attribute TLVs

Link Attribute TLVs are TLVs that may be encoded in the BGP-LS attribute with a Link NLRI. Each 'Link Attribute' is a Type/Length/Value (TLV) triplet formatted as defined in Section 4.1. The format and semantics of the Value fields in some Link Attribute TLVs correspond to the format and semantics of the Value fields in IS-IS Extended IS Reachability sub-TLVs, defined in [RFC5305] and [RFC5307]. Other Link Attribute TLVs are defined in this document. Although the encodings for Link Attribute TLVs were originally defined for IS-IS, the TLVs can carry data sourced by either IS-IS or OSPF.

The following Link Attribute TLVs are defined for the BGP-LS Attribute associated with a Link NLRI:

TLV Code Point	Description	IS-IS TLV/Sub-TLV	Reference (RFC/Section)
1028	IPv4 Router-ID of Local Node	134/---	[RFC5305] / 4.3
1029	IPv6 Router-ID of Local Node	140/---	[RFC6119] / 4.1
1030	IPv4 Router-ID of Remote Node	134/---	[RFC5305] / 4.3
1031	IPv6 Router-ID of Remote Node	140/---	[RFC6119] / 4.1
1088	Administrative group (color)	22/3	[RFC5305] / 3.1
1089	Maximum link bandwidth	22/9	[RFC5305] / 3.4
1090	Max. reservable link bandwidth	22/10	[RFC5305] / 3.5
1091	Unreserved bandwidth	22/11	[RFC5305] / 3.6
1092	TE Default Metric	22/18	Section 4.3.2.3
1093	Link Protection Type	22/20	[RFC5307] / 1.2
1094	MPLS Protocol Mask	---	Section 4.3.2.2
1095	IGP Metric	---	Section 4.3.2.4
1096	Shared Risk Link Group	---	Section 4.3.2.5
1097	Opaque Link Attribute	---	Section 4.3.2.6
1098	Link Name	---	Section 4.3.2.7

Table 8: Link Attribute TLVs

4.3.2.1. IPv4/IPv6 Router-ID TLVs

The local/remote IPv4/IPv6 Router-ID TLVs are used to describe auxiliary Router-IDs that the IGP might be using, e.g., for TE purposes. All auxiliary Router-IDs of both the local and the remote node MUST be included in the link attribute of each Link NLRI. If there is more than one auxiliary Router-ID of a given type, then multiple TLVs are used to encode them.

4.3.2.2. MPLS Protocol Mask TLV

The MPLS Protocol Mask TLV carries a bit mask describing which MPLS signaling protocols are enabled. The length of this TLV is 1. The value is a bit array of 8 flags, where each bit represents an MPLS Protocol capability.

Generation of the MPLS Protocol Mask TLV is only valid for and SHOULD only be used with originators that have local link insight, for example, the Protocol-IDs 'Static configuration' or 'Direct' as per Table 2. The MPLS Protocol Mask TLV MUST NOT be included in NLRIs with the other Protocol-IDs listed in Table 2.

```
     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            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |L|R|  Reserved |
    +-+-+-+-+-+-+-+-+
```

Figure 19: MPLS Protocol Mask TLV

The following bits are defined:

Bit	Description	Reference
'L'	Label Distribution Protocol (LDP)	[RFC5036]
'R'	Extension to RSVP for LSP Tunnels (RSVP-TE)	[RFC3209]

Table 9: MPLS Protocol Mask TLV Codes

4.3.2.3. TE Default Metric TLV

The TE Default Metric TLV carries the Traffic Engineering metric for this link. The length of this TLV is fixed at 4 octets. If a source protocol uses a metric width of less than 32 bits, then the high-order bits of this field MUST be padded with zero.

```
     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            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                    TE Default Link Metric                     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 20: TE Default Metric TLV Format

4.3.2.4. IGP Metric TLV

The IGP Metric TLV carries the metric for this link. The length of this TLV is variable, depending on the metric width of the underlying protocol. IS-IS small metrics have a length of 1 octet. Since the ISIS small metrics are of 6 bit size, the two most significant bits MUST be set to 0 and MUST be ignored by receiver. OSPF link metrics have a length of 2 octets. IS-IS wide metrics have a length of 3 octets.

```
      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            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     //      IGP Link Metric (variable length)      //
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 21: IGP Metric TLV Format

4.3.2.5. Shared Risk Link Group TLV

The Shared Risk Link Group (SRLG) TLV carries the Shared Risk Link Group information (see Section 2.3 ("Shared Risk Link Group Information") of [RFC4202]). It contains a data structure consisting of a (variable) list of SRLG values, where each element in the list has 4 octets, as shown in Figure 22. The length of this TLV is 4 * (number of SRLG values).

```
      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            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                  Shared Risk Link Group Value                 |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     //                         ............                        //
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                  Shared Risk Link Group Value                 |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 22: Shared Risk Link Group TLV Format

The SRLG TLV for OSPF-TE is defined in [RFC4203]. In IS-IS, the SRLG information is carried in two different TLVs: the IPv4 (SRLG) TLV (Type 138) defined in [RFC5307] and the IPv6 SRLG TLV (Type 139) defined in [RFC6119]. In Link-State NLRI, both IPv4 and IPv6 SRLG information are carried in a single TLV.

4.3.2.6. Opaque Link Attribute TLV

The Opaque Link Attribute TLV is an envelope that transparently carries optional Link Attribute TLVs advertised by a router. An originating router shall use this TLV for encoding information specific to the protocol advertised in the NLRI header Protocol-ID field or new protocol extensions to the protocol as advertised in the NLRI header Protocol-ID field for which there is no protocol-neutral representation in the BGP Link-State NLRI. The primary use of the Opaque Link Attribute TLV is to bridge the document lag between, e.g., a new IGP link-state attribute being defined and the 'protocol-neutral' BGP-LS extensions being published.

In the case of OSPFv2, this TLV MAY be used to advertise information carried using the TLVs in the "OSPFv2 Extended Link Opaque LSA TLVs" registry [RFC7684] under the IANA OSPFv2 Parameters registry. In the case of OSPFv3, this TLV MAY be used to advertise information carried using the TLVs in the "OSPFv3 Extended-LSA Sub-TLVs" registry [RFC8362] under the IANA OSPFv3 Parameters registry.

```
     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            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //                Opaque link attributes (variable)            //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 23: Opaque Link Attribute TLV Format

4.3.2.7. Link Name TLV

The Link Name TLV is optional. The Value field identifies the symbolic name of the router link. This symbolic name can be the FQDN for the link, it can be a subset of the FQDN, or it can be any string operators want to use for the link. The use of FQDN or a subset of it is strongly RECOMMENDED. The maximum length of the Link Name TLV is 255 octets.

The Value field is encoded in 7-bit ASCII. If a user interface for configuring or displaying this field permits Unicode characters, that user interface is responsible for applying the ToASCII and/or ToUnicode algorithm as described in [RFC5890] to achieve the correct format for transmission or display.

How a router derives and injects link names is outside of the scope of this document.

```
     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            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //                     Link Name (variable)                    //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 24: Link Name TLV Format

4.3.3. Prefix Attribute TLVs

Prefixes are learned from the IGP topology (IS-IS or OSPF) with a set of IGP attributes (such as metric, route tags, etc.) that are advertised in the BGP-LS Attribute with Prefix NLRI types 3 and 4.

The following Prefix Attribute TLVs are defined for the BGP-LS Attribute associated with a Prefix NLRI:

TLV Code Point	Description	Length	Reference
1152	IGP Flags	1	Section 4.3.3.1
1153	IGP Route Tag	4*n	[RFC5130]
1154	IGP Extended Route Tag	8*n	[RFC5130]
1155	Prefix Metric	4	[RFC5305]
1156	OSPF Forwarding Address	4	[RFC2328]
1157	Opaque Prefix Attribute	variable	Section 4.3.3.6

Table 10: Prefix Attribute TLVs

4.3.3.1. IGP Flags TLV

The IGP Flags TLV contains one octet of IS-IS and OSPF flags and bits originally assigned to the prefix. The IGP Flags TLV is encoded 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
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |              Type             |             Length            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |D|N|L|P|       |
     +-+-+-+-+-+-+-+-+
```

Figure 25: IGP Flag TLV Format

The Value field contains bits defined according to the table below:

Bit	Description	Reference
'D'	IS-IS Up/Down Bit	[RFC5305]
'N'	OSPF "no unicast" Bit	[RFC5340]
'L'	OSPF "local address" Bit	[RFC5340]
'P'	OSPF "propagate NSSA" Bit	[RFC5340]

Table 11: IGP Flag Bits Definitions

4.3.3.2. IGP Route Tag TLV

The IGP Route Tag TLV carries original IGP Tags (IS-IS [RFC5130] or OSPF) of the prefix and is encoded 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
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |              Type             |             Length            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     //                    Route Tags (one or more)                  //
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 26: IGP Route Tag TLV Format

Length is a multiple of 4.

The Value field contains one or more Route Tags as learned in the IGP topology.

4.3.3.3. Extended IGP Route Tag TLV

The Extended IGP Route Tag TLV carries IS-IS Extended Route Tags of the prefix [RFC5130] and is encoded 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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Type             |             Length            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //                Extended Route Tag (one or more)             //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 27: Extended IGP Route Tag TLV Format

Length is a multiple of 8.

The Extended Route Tag field contains one or more Extended Route Tags as learned in the IGP topology.

4.3.3.4. Prefix Metric TLV

The Prefix Metric TLV is an optional attribute and may only appear once. If present, it carries the metric of the prefix as known in the IGP topology as described in Section 4 of [RFC5305] (and therefore represents the reachability cost to the prefix). If not present, it means that the prefix is advertised without any reachability.

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

Figure 28: Prefix Metric TLV Format

Length is 4.

4.3.3.5. OSPF Forwarding Address TLV

The OSPF Forwarding Address TLV [RFC2328] [RFC5340] carries the OSPF forwarding address as known in the original OSPF advertisement. Forwarding address can be either IPv4 or IPv6.

```
     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            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //                Forwarding Address (variable)                //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 29: OSPF Forwarding Address TLV Format

Length is 4 for an IPv4 forwarding address, and 16 for an IPv6 forwarding address.

4.3.3.6. Opaque Prefix Attribute TLV

The Opaque Prefix Attribute TLV is an envelope that transparently carries optional Prefix Attribute TLVs advertised by a router. An originating router shall use this TLV for encoding information specific to the protocol advertised in the NLRI header Protocol-ID field or new protocol extensions to the protocol as advertised in the NLRI header Protocol-ID field for which there is no protocol-neutral representation in the BGP Link-State NLRI. The primary use of the Opaque Prefix Attribute TLV is to bridge the document lag between, e.g., a new IGP link-state attribute being defined and the protocol-neutral BGP-LS extensions being published.

In the case of OSPFv2, this TLV MAY be used to advertise information carried using the TLVs in the "OSPFv2 Extended Prefix Opaque LSA TLVs" registry [RFC7684] under the IANA OSPFv2 Parameters registry. In the case of OSPFv3, this TLV MAY be used to advertise information carried using the TLVs in the "OSPFv3 Extended-LSA Sub-TLVs" registry [RFC8362] under the IANA OSPFv3 Parameters registry.

The format of the TLV 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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Type             |             Length            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //              Opaque Prefix Attributes  (variable)           //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 30: Opaque Prefix Attribute TLV Format

Type is as specified in Table 10. Length is variable.

4.4. Private Use

TLVs for Vendor Private use are supported using the code point range reserved as indicated in Section 6. For such TLV use in the NLRI or BGP-LS Attribute, the format as described in Section 4.1 is to be used and a 4 octet field MUST be included as the first field in the value to carry the Enterprise Code. For a private use NLRI Type, a 4 octet field MUST be included as the first field in the NLRI immediately following the Total NLRI Length field of the Link-State NLRI format as described in Section 4.2 to carry the Enterprise Code. The Enterprise Codes are listed at <http://www.iana.org/assignments/enterprise-numbers>. This enables use vendor specific extensions without conflicts.

Multiple instances of private-use TLVs MAY appear in the BGP-LS Attribute.

4.5. BGP Next-Hop Information

BGP link-state information for both IPv4 and IPv6 networks can be carried over either an IPv4 BGP session or an IPv6 BGP session. If an IPv4 BGP session is used, then the next hop in the MP_REACH_NLRI SHOULD be an IPv4 address. Similarly, if an IPv6 BGP session is used, then the next hop in the MP_REACH_NLRI SHOULD be an IPv6 address. Usually, the next hop will be set to the local endpoint address of the BGP session. The next-hop address MUST be encoded as described in [RFC4760]. The Length field of the next-hop address will specify the next-hop address family. If the next-hop length is 4, then the next hop is an IPv4 address; if the next-hop length is 16, then it is a global IPv6 address; and if the next-hop length is 32, then there is one global IPv6 address followed by a link-local IPv6 address. The link-local IPv6 address should be used as described in [RFC2545]. For VPN Subsequent Address Family Identifier (SAFI), as per custom, an 8-byte Route Distinguisher set to all zero is prepended to the next hop.

The BGP Next Hop attribute is used by each BGP-LS speaker to validate the NLRI it receives. In case identical NLRIs are sourced by multiple BGP-LS Producers, the BGP Next Hop attribute is used to tiebreak as per the standard BGP path decision process. This specification doesn’t mandate any rule regarding the rewrite of the BGP Next Hop attribute.

4.6. Inter-AS Links

The main source of TE information is the IGP, which is not active on inter-AS links. In some cases, the IGP may have information of inter-AS links [RFC5392] [RFC5316]. In other cases, an implementation SHOULD provide a means to inject inter-AS links into BGP-LS. The exact mechanism used to advertise the inter-AS links is outside the scope of this document.

4.7. OSPF Virtual Links and Sham Links

In an OSPF [RFC2328] [RFC5340] network, virtual links serve to connect physically separate components of the backbone to establish/maintain continuity of the backbone area. While virtual links are modeled as point-to-point unnumbered links in the OSPF topology, their characteristics and purpose are different from other types of links in the OSPF topology. They are advertised using a distinct "virtual link" type in OSPF LSAs. The mechanism for advertisement of OSPF virtual links via BGP-LS is outside the scope of this document.

In an OSPF network, sham links [RFC4577] [RFC6565] are used to provide an intra-area connectivity between VRFs on PE routers over the VPN provider's network. These links are advertised in OSPF as a point-to-point unnumbered links and represent connectivity over a service provider network using encapsulation mechanisms like MPLS. As such, the mechanism for advertisement of OSPF sham links follow the same procedures as other point-to-point unnumbered links as described previously in this document.

4.8. OSPFv2 Type 4 Summary LSA & OSPFv3 Inter-Area Router LSA

OSPFv2 [RFC2328] defines the Type 4 Summary LSA and OSPFv3 [RFC5340] the Inter-area-router-LSA for an Area Border Router (ABR) to advertise reachability to an AS Border Router (ASBR) that is external to the area yet internal to the AS. The nature of information advertised by OSPF using this type of LSA does not map to either a node or a link or a prefix as discussed in this document. Therefore, the mechanism for advertisement of the information carried by these LSAs is outside the scope of this document.

4.9. Handling of Unreachable IGP Nodes

The origination and propagation of IGP link-state information via BGP needs to provide a consistent and true view of the topology of the IGP domain. BGP-LS provides an abstraction of the protocol specifics and BGP-LS Consumers may be varied types of applications. While the information propagated via BGP-LS from a link-state routing protocol is sourced from that protocol's LSDB, it does not serve as a true reflection of the originating router's LSDB since it does not include the LSA/LSP sequence number information. The sequence numbers are not included since a single NLRI update may be put together with information that is coming from multiple LSAs/LSPs.

Consider an OSPF network as shown in Figure 31, where R2 and R3 are the BGP-LS Producers and also the OSPF Area Border Routers (ABRs). The link between R2 and R3 is in area 0 while the other links shown are in area 1.

A BGP-LS Consumer talks to a BGP route-reflector (RR) R0 which is aggregating the BGP-LS feed from the BGP-LS Producers R2 and R3. Here R2 and R3 provide a redundant topology feed via BGP-LS to R0. Normally, R0 would receive two identical copies of all the Link-State NLRIs from both R2 and R3 and it would pick one of them (say R2) based on the standard BGP best path decision process.

```
                      Consumer
                         ^
                         |
                         R0
                 (BGP Route Reflector)
                      /      \
                     /        \
            a1      /   a0     \    a1
         R1 ------ R2 -------- R3 ------ R4
     a1  |                               |  a1
         |                               |
         R5 ---------------------------- R6
                        a1
```

Figure 31: Incorrect Reporting due to BGP Path Selection

Consider a scenario where the link between R5 and R6 is lost (thereby partitioning the area 1) and its impact on the OSPF LSDB at R2 and R3.

Now, R5 will remove the link 5-6 from its Router LSA and this updated LSA is available at R2. R2 also has a stale copy of R6's Router LSA which still has the link 6-5 in it. Based on this view in its LSDB, R2 will advertise only the half-link 6-5 that it derives from R6's stale Router LSA.

At the same time, R6 has removed the link 6-5 from its Router LSA and this updated LSA is available at R3. Similarly, R3 also has a stale copy of R5's Router LSA having the link 5-6 in it. Based on it's LSDB, R3 will advertise only the half-link 5-6 that it has derived from R5's stale Router LSA.

Now, the BGP-LS Consumer receives both the Link NLRIs corresponding to the half-links from R2 and R3 via R0. When viewed together, it would not detect or realize that the area 1 is actually partitioned.

Also if R2 continues to report Link-State NLRIs corresponding to the stale copy of Router LSA of R4 and R6 nodes then R0 would prefer them over the valid Link-State NLRIs for R4 and R6 that it is receiving from R3 based on its BGP decision process. This would result in the BGP-LS Consumer getting stale and inaccurate topology information. This problems scenario is avoided if R2 were to not advertise the link-state information corresponding to R4 and R6 and if R3 were to not advertise similarly for R1 and R5.

A BGP-LS Producer SHOULD withdraw all link-state objects advertised by it in BGP when the node that originated its corresponding LSP/LSAs is determined to have become unreachable in the IGP. An implementation MAY continue to advertise link-state objects corresponding to unreachable nodes in a deployment use-case where the BGP-LS Consumer is interested in receiving a topology feed corresponding to a complete IGP LSDB view. In such deployments, it is expected that the problem described above is mitigated by the BGP-LS Consumer via appropriate handling of such a topology feed in addition to the use of either a direct BGP peering with the producer nodes or mechanisms such as [RFC7911] when using RR. Details of these mechanisms are outside the scope of this draft.

If the BGP-LS Producer does withdraw link-state objects associated with an IGP node based on failure of reachability check for that node, then it MUST re-advertise those link-state objects after that node becomes reachable again in the IGP domain.

4.10. Router-ID Anchoring Example: ISO Pseudonode

Encoding of a broadcast LAN in IS-IS provides a good example of how Router-IDs are encoded. Consider Figure 32. This represents a Broadcast LAN between a pair of routers. The "real" (non-pseudonode) routers have both an IPv4 Router-ID and IS-IS Node-ID. The pseudonode does not have an IPv4 Router-ID. Node1 is the DIS for the LAN. Two unidirectional links (Node1, Pseudonode1) and (Pseudonode1, Node2) are being generated.

The Link NLRI of (Node1, Pseudonode1) is encoded as follows. The IGP Router-ID TLV of the local Node Descriptor is 6 octets long and contains the ISO-ID of Node1, 1920.0000.2001. The IGP Router-ID TLV of the remote Node Descriptor is 7 octets long and contains the ISO-ID of Pseudonode1, 1920.0000.2001.02. The BGP-LS attribute of this link contains one local IPv4 Router-ID TLV (TLV type 1028) containing 192.0.2.1, the IPv4 Router-ID of Node1.

The Link NLRI of (Pseudonode1, Node2) is encoded as follows. The IGP Router-ID TLV of the local Node Descriptor is 7 octets long and contains the ISO-ID of Pseudonode1, 1920.0000.2001.02. The IGP

Router-ID TLV of the remote Node Descriptor is 6 octets long and contains the ISO-ID of Node2, 1920.0000.2002. The BGP-LS attribute of this link contains one remote IPv4 Router-ID TLV (TLV type 1030) containing 192.0.2.2, the IPv4 Router-ID of Node2.

```
+-----------------+    +-----------------+    +-----------------+
|      Node1      |    |   Pseudonode1   |    |      Node2      |
|1920.0000.2001.00|--->|1920.0000.2001.02|--->|1920.0000.2002.00|
|    192.0.2.1    |    |                 |    |    192.0.2.2    |
+-----------------+    +-----------------+    +-----------------+
```

Figure 32: IS-IS Pseudonodes

4.11. Router-ID Anchoring Example: OSPF Pseudonode

Encoding of a broadcast LAN in OSPF provides a good example of how Router-IDs and local Interface IPs are encoded. Consider Figure 33. This represents a Broadcast LAN between a pair of routers. The "real" (non-pseudonode) routers have both an IPv4 Router-ID and an Area Identifier. The pseudonode does have an IPv4 Router-ID, an IPv4 Interface Address (for disambiguation), and an OSPF Area. Node1 is the DR for the LAN; hence, its local IP address 10.1.1.1 is used as both the Router-ID and Interface IP for the pseudonode keys. Two unidirectional links, (Node1, Pseudonode1) and (Pseudonode1, Node2), are being generated.

The Link NLRI of (Node1, Pseudonode1) is encoded as follows:

o Local Node Descriptor

TLV #515: IGP Router-ID: 11.11.11.11

TLV #514: OSPF Area-ID: ID:0.0.0.0

o Remote Node Descriptor

TLV #515: IGP Router-ID: 11.11.11.11:10.1.1.1

TLV #514: OSPF Area-ID: ID:0.0.0.0

The Link NLRI of (Pseudonode1, Node2) is encoded as follows:

o Local Node Descriptor

TLV #515: IGP Router-ID: 11.11.11.11:10.1.1.1

TLV #514: OSPF Area-ID: ID:0.0.0.0

- Remote Node Descriptor

  TLV #515: IGP Router-ID: 33.33.33.34

  TLV #514: OSPF Area-ID: ID:0.0.0.0

```
         10.1.1.1/24              10.1.1.2/24
+-------------+    +-------------+    +-------------+
|    Node1    |    | Pseudonode1 |    |    Node2    |
| 11.11.11.11 |--->| 11.11.11.11 |--->| 33.33.33.34 |
|             |    |   10.1.1.1  |    |             |
|    Area 0   |    |   Area 0    |    |    Area 0   |
+-------------+    +-------------+    +-------------+
```

Figure 33: OSPF Pseudonodes

The LAN subnet 10.1.1.0/24 is not included in the Router LSA of Node1 or Node2. The Network LSA for this LAN advertised by the DR Node1 contains the subnet mask for the LAN along with the DR address. A Prefix NLRI corresponding to the LAN subnet is advertised with the Pseudonode1 used as the Local node using the DR address and the subnet mask from the Network LSA.

4.12. Router-ID Anchoring Example: OSPFv2 to IS-IS Migration

Graceful migration from one IGP to another requires coordinated operation of both protocols during the migration period. Such a coordination requires identifying a given physical link in both IGPs. The IPv4 Router-ID provides that "glue", which is present in the Node Descriptors of the OSPF Link NLRI and in the link attribute of the IS-IS Link NLRI.

Consider a point-to-point link between two routers, A and B, that initially were OSPFv2-only routers and then IS-IS is enabled on them. Node A has IPv4 Router-ID and ISO-ID; node B has IPv4 Router-ID, IPv6 Router-ID, and ISO-ID. Each protocol generates one Link NLRI for the link (A, B), both of which are carried by BGP-LS. The OSPFv2 Link NLRI for the link is encoded with the IPv4 Router-ID of nodes A and B in the local and remote Node Descriptors, respectively. The IS-IS Link NLRI for the link is encoded with the ISO-ID of nodes A and B in the local and remote Node Descriptors, respectively. In addition, the BGP-LS attribute of the IS-IS Link NLRI contains the TLV type 1028 containing the IPv4 Router-ID of node A, TLV type 1030 containing the IPv4 Router-ID of node B, and TLV type 1031 containing the IPv6 Router-ID of node B. In this case, by using IPv4 Router-ID, the link (A, B) can be identified in both the IS-IS and OSPF protocol.

5. Link to Path Aggregation

Distribution of all links available in the global Internet is certainly possible; however, it not desirable from a scaling and privacy point of view. Therefore, an implementation may support a link to path aggregation. Rather than advertising all specific links of a domain, an ASBR may advertise an "aggregate link" between a non-adjacent pair of nodes. The "aggregate link" represents the aggregated set of link properties between a pair of non-adjacent nodes. The actual methods to compute the path properties (of bandwidth, metric, etc.) are outside the scope of this document. The decision whether to advertise all specific links or aggregated links is an operator’s policy choice. To highlight the varying levels of exposure, the following deployment examples are discussed.

5.1. Example: No Link Aggregation

Consider Figure 34. Both AS1 and AS2 operators want to protect their inter-AS {R1, R3}, {R2, R4} links using RSVP-FRR LSPs. If R1 wants to compute its link-protection LSP to R3, it needs to "see" an alternate path to R3. Therefore, the AS2 operator exposes its topology. All BGP-TE-enabled routers in AS1 "see" the full topology of AS2 and therefore can compute a backup path. Note that the computing router decides if the direct link between {R3, R4} or the {R4, R5, R3} path is used.

```
          AS1   :   AS2
                :
           R1-------R3
            |   :   | \
            |   :   |  R5
            |   :   | /
           R2-------R4
                :
                :
```

Figure 34: No Link Aggregation

5.2. Example: ASBR to ASBR Path Aggregation

The brief difference between the "no-link aggregation" example and this example is that no specific link gets exposed. Consider Figure 35. The only link that gets advertised by AS2 is an "aggregate" link between R3 and R4. This is enough to tell AS1 that there is a backup path. However, the actual links being used are hidden from the topology.

```
          AS1   :   AS2
                :
           R1-------R3
            |   :   |
            |   :   |
            |   :   |
           R2-------R4
                :
                :
```

Figure 35: ASBR Link Aggregation

5.3. Example: Multi-AS Path Aggregation

Service providers in control of multiple ASes may even decide to not expose their internal inter-AS links. Consider Figure 36. AS3 is modeled as a single node that connects to the border routers of the aggregated domain.

```
          AS1   :   AS2   :   AS3
                :         :
           R1-------R3-----
            |   :         : \
            |   :         :   vR0
            |   :         : /
           R2-------R4-----
                :         :
                :         :
```

Figure 36: Multi-AS Aggregation

6. IANA Considerations

IANA has assigned address family number 16388 (BGP-LS) in the "Address Family Numbers" registry with [RFC7752] as a reference.

IANA has assigned SAFI values 71 (BGP-LS) and 72 (BGP-LS-VPN) in the "SAFI Values" sub-registry under the "Subsequent Address Family Identifiers (SAFI) Parameters" registry with [RFC7752] as a reference.

IANA has assigned value 29 (BGP-LS Attribute) in the "BGP Path Attributes" sub-registry under the "Border Gateway Protocol (BGP) Parameters" registry with [RFC7752] as a reference.

IANA has created a new "Border Gateway Protocol - Link State (BGP-LS) Parameters" registry at <https://www.iana.org/assignments/bgp-ls-parameters>.

6.1. BGP-LS Registries

All of the registries listed in the following sub-sections are BGP-LS specific and are accessible under this registry.

6.1.1. BGP-LS NLRI Types Registry

The "BGP-LS NLRI Types" registry has been setup for assignment for the two octet sized code-points for BGP-LS NLRI types and populated with the values shown below:

Type	NLRI Type	Reference
0	Reserved	[RFC7752][This document]
1	Node NLRI	[RFC7752][This document]
2	Link NLRI	[RFC7752][This document]
3	IPv4 Topology Prefix NLRI	[RFC7752][This document]
4	IPv6 Topology Prefix NLRI	[RFC7752][This document]
65000-65535	Private Use	[This document]

Allocations within the registry under the "Expert Review" policy require documentation of the proposed use of the allocated value and approval by the Designated Expert assigned by the IESG (see [RFC8126]).

6.1.2. BGP-LS Protocol-IDs Registry

The "BGP-LS Protocol-IDs" registry has been setup for assignment for the one octet sized code-points for BGP-LS Protocol-IDs and populated with the values shown below:

Protocol-ID	NLRI information source protocol	Reference
0	Reserved	[RFC7752][This document]
1	IS-IS Level 1	[RFC7752][This document]
2	IS-IS Level 2	[RFC7752][This document]
3	OSPFv2	[RFC7752][This document]
4	Direct	[RFC7752][This document]
5	Static configuration	[RFC7752][This document]
6	OSPFv3	[RFC7752][This document]
200-255	Private Use	[This document]

Allocations within the registry under the "Expert Review" policy require documentation of the proposed use of the allocated value and approval by the Designated Expert assigned by the IESG (see [RFC8126]).

6.1.3. BGP-LS Well-Known Instance-IDs Registry

The "BGP-LS Well-Known Instance-IDs" registry that was setup via [RFC7752] is no longer required. It may be retained as deprecated and no further assignments be made from it.

6.1.4. BGP-LS Node Flags Registry

The "BGP-LS Node Flags" registry is requested to be created for the 1 octet sized flags field of the Node Flag Bits TLV (1024) and populated with the initial values shown below:

Bit	Description	Reference
0	Overload Bit (O-bit)	[RFC7752][This document]
1	Attached Bit (A-bit)	[RFC7752][This document]
2	External Bit (E-bit)	[RFC7752][This document]
3	ABR Bit (B-bit)	[RFC7752][This document]
4	Router Bit (R-bit)	[RFC7752][This document]
5	V6 Bit (V-bit)	[RFC7752][This document]
6-7	Unassigned	

Allocations within the registry under the "Specification Required" policy (see [RFC8126]).

6.1.5. BGP-LS MPLS Protocol Mask Registry

The "BGP-LS MPLS Protocol Mask" registry is requested to be created for the 1 octet sized flags field of the MPLS Protocol Mask TLV (1094) and populated with the initial values shown below:

Bit	Description	Reference
0	Label Distribution Protocol (L-bit)	[RFC7752][This document]
1	Extension to RSVP for LSP Tunnels (R-bit)	[RFC7752][This document]
2-7	Unassigned	

Allocations within the registry under the "Specification Required" policy (see [RFC8126]).

6.1.6. BGP-LS IGP Prefix Flags Registry

The "BGP-LS IGP Prefix Flags" registry is requested to be created for the 1 octet sized flags field of the IGP Flags TLV (1152) and populated with the initial values shown below:

Bit	Description	Reference
0	IS-IS Up/Down Bit (D-bit)	[RFC7752][This document]
1	OSPF "no unicast" Bit (N-bit)	[RFC7752][This document]
2	OSPF "local address" Bit (L-bit)	[RFC7752][This document]
3	OSPF "propagate NSSA" Bit (P-bit)	[RFC7752][This document]
4-7	Unassigned	

Allocations within the registry under the "Specification Required" policy (see [RFC8126]).

6.1.7. BGP-LS TLVs Registry

The "BGP-LS Node Descriptor, Link Descriptor, Prefix Descriptor, and Attribute TLVs" registry was setup via [RFC7752]. Values 0-255 are reserved. Values 256-65535 will be used for code points. The range 65000-65535 is for Private Use. The registry has been populated with the values shown in Table 12. Allocations within the registry under the "Expert Review" policy require documentation of the proposed use of the allocated value and approval by the Designated Expert assigned by the IESG (see [RFC8126]).

6.2. Guidance for Designated Experts

In all cases of review by the Designated Expert (DE) described here, the DE is expected to ascertain the existence of suitable documentation (a specification) as described in [RFC8126]. The DE is also expected to check the clarity of purpose and use of the requested code points. Additionally, the DE must verify that any request for one of these code points has been made available for review and comment within the IETF: the DE will post the request to the IDR Working Group mailing list (or a successor mailing list designated by the IESG). If the request comes from within the IETF, it should be documented in an Internet-Draft. Lastly, the DE must ensure that any other request for a code point does not conflict with work that is active or already published within the IETF.

7. Manageability Considerations

This section is structured as recommended in [RFC5706].

7.1. Operational Considerations

7.1.1. Operations

Existing BGP operational procedures apply. No new operation procedures are defined in this document. It is noted that the NLRI information present in this document carries purely application-level data that has no immediate impact on the corresponding forwarding state computed by BGP. As such, any churn in reachability information has a different impact than regular BGP updates, which need to change the forwarding state for an entire router. It is expected that the distribution of this NLRI SHOULD be handled by dedicated route reflectors in most deployments providing a level of isolation and fault containment between different NLRI types. In the event of dedicated route reflectors not being available, other alternate mechanisms like separation of BGP instances or separate BGP sessions (e.g. using different addresses for peering) for Link-State information distribution SHOULD be used.

7.1.2. Installation and Initial Setup

Configuration parameters defined in Section 7.2.3 SHOULD be initialized to the following default values:

- The Link-State NLRI capability is turned off for all neighbors.

- The maximum rate at which Link-State NLRIs will be advertised/ withdrawn from neighbors is set to 200 updates per second.

7.1.3. Migration Path

The proposed extension is only activated between BGP peers after capability negotiation. Moreover, the extensions can be turned on/ off on an individual peer basis (see Section 7.2.3), so the extension can be gradually rolled out in the network.

7.1.4. Requirements on Other Protocols and Functional Components

The protocol extension defined in this document does not put new requirements on other protocols or functional components.

7.1.5. Impact on Network Operation

Frequency of Link-State NLRI updates could interfere with regular BGP prefix distribution. A network operator MAY use a dedicated Route-Reflector infrastructure to distribute Link-State NLRIs.

Distribution of Link-State NLRIs SHOULD be limited to a single admin domain, which can consist of multiple areas within an AS or multiple ASes.

7.1.6. Verifying Correct Operation

Existing BGP procedures apply. In addition, an implementation SHOULD allow an operator to:

- List neighbors with whom the speaker is exchanging Link-State NLRIs.

7.2. Management Considerations

7.2.1. Management Information

The IDR working group has documented and continues to document parts of the Management Information Base and YANG models for managing and monitoring BGP speakers and the sessions between them. It is currently believed that the BGP session running BGP-LS is not substantially different from any other BGP session and can be managed using the same data models.

7.2.2. Fault Management

This section describes the fault management actions, as described in [RFC7606] , that are to be performed for handling of BGP update messages for BGP-LS.

A Link-State NLRI MUST NOT be considered as malformed or invalid based on the inclusion/exclusion of TLVs or contents of the TLV fields (i.e. semantic errors), as described in Section 4.1 and Section 4.2.

A BGP-LS Speaker MUST perform the following syntactic validation of the Link-State NLRI to determine if it is malformed.

- Does the sum of all TLVs found in the BGP MP_REACH_NLRI attribute correspond to the BGP MP_REACH_NLRI length?

- Does the sum of all TLVs found in the BGP MP_UNREACH_NLRI attribute correspond to the BGP MP_UNREACH_NLRI length?

- Does the sum of all TLVs found in a Link-State NLRI correspond to the Total NLRI Length field of all its Descriptors?

- Is the length of the TLVs and, when the TLV is recognized then, its sub-TLVs in the NLRI valid?

- Has the syntactic correctness of the NLRI fields been verified as per [RFC7606]?

- Has the rule regarding ordering of TLVs been followed as described in Section 4.1?

When the error determined allows for the router to skip the malformed NLRI(s) and continue processing of the rest of the update message (e.g. when the TLV ordering rule is violated), then it MUST handle such malformed NLRIs as 'Treat-as-withdraw'. In other cases, where the error in the NLRI encoding results in the inability to process the BGP update message (e.g. length related encoding errors), then the router SHOULD handle such malformed NLRIs as 'AFI/SAFI disable' when other AFI/SAFI besides BGP-LS are being advertised over the same session. Alternately, the router MUST perform 'session reset' when the session is only being used for BGP-LS or when it 'AFI/SAFI disable' action is not possible.

A BGP-LS Attribute MUST NOT be considered as malformed or invalid based on the inclusion/exclusion of TLVs or contents of the TLV fields (i.e. semantic errors), as described in Section 4.1 and Section 4.3.

A BGP-LS Speaker MUST perform the following syntactic validation of the BGP-LS Attribute to determine if it is malformed.

- Does the sum of all TLVs found in the BGP-LS Attribute correspond to the BGP-LS Attribute length?

- Has the syntactic correctness of the Attributes (including BGP-LS Attribute) been verified as per [RFC7606]?

- Is the length of each TLV and, when the TLV is recognized then, its sub-TLVs in the BGP-LS Attribute valid?

When the error determined allows for the router to skip the malformed BGP-LS Attribute and continue processing of the rest of the update message (e.g. when the BGP-LS Attribute length and the total Path Attribute Length are correct but some TLV/sub-TLV length within the BGP-LS Attribute is invalid), then it MUST handle such malformed BGP-LS Attribute as 'Attribute Discard'. In other cases, where the error in the BGP-LS Attribute encoding results in the inability to process the BGP update message then the handling is the same as described above for the malformed NLRI.

Note that the 'Attribute Discard' action results in the loss of all TLVs in the BGP-LS Attribute and not the removal of a specific malformed TLV. The removal of specific malformed TLVs may give a wrong indication to a BGP-LS Consumer of that specific information being deleted or not available.

When a BGP Speaker receives an update message with Link-State NLRI(s) in the MP_REACH_NLRI but without the BGP-LS Attribute, it is most likely an indication that a BGP Speaker preceding it has performed the 'Attribute Discard' fault handling. An implementation SHOULD preserve and propagate the Link-State NLRIs in such an update message so that the BGP-LS Consumers can detect the loss of link-state information for that object and not assume its deletion/withdraw. This also makes it possible for a network operator to trace back to the BGP-LS Propagator which actually detected a fault with the BGP-LS Attribute.

An implementation SHOULD log an error for any errors found during syntax validation for further analysis.

A BGP-LS Propagator SHOULD NOT perform semantic validation of the Link-State NLRI or the BGP-LS Attribute to determine if it is malformed or invalid. Some types of semantic validation that are not to be performed by a BGP-LS Propagator are as follows (and this is not to be considered as an exhaustive list):

- is a mandatory TLV present or not?

- is the length of a fixed length TLV correct or the length of a variable length TLV a valid/permissible?

- are the values of TLV fields valid or permissible?

- are the inclusion and use of TLVs/sub-TLVs with specific Link-State NLRI types valid?

Each TLV MAY indicate the valid and permissible values and their semantics that can to be used only by a BGP-LS Consumer for its semantic validation. However, the handling of any errors may be specific to the particular application and outside the scope of this document. A BGP-LS Consumer should ignore unrecognized and unexpected TLV types in both the NLRI and BGP-LS Attribute portions and not consider their presence as an error.

7.2.3. Configuration Management

An implementation SHOULD allow the operator to specify neighbors to which Link-State NLRIs will be advertised and from which Link-State NLRIs will be accepted.

An implementation SHOULD allow the operator to specify the maximum rate at which Link-State NLRIs will be advertised/withdrawn from neighbors.

An implementation SHOULD allow the operator to specify the maximum number of Link-State NLRIs stored in a router’s Routing Information Base (RIB).

An implementation SHOULD allow the operator to create abstracted topologies that are advertised to neighbors and create different abstractions for different neighbors.

An implementation SHOULD allow the operator to configure a 64-bit Instance-ID.

An implementation SHOULD allow the operator to configure ASN and BGP-LS identifiers (refer Section 4.2.1.4).

An implementation SHOULD allow the operator to configure the maximum size of the BGP-LS Attribute that may be used on a BGP-LS Producer.

7.2.4. Accounting Management

Not Applicable.

7.2.5. Performance Management

An implementation SHOULD provide the following statistics:

- o Total number of Link-State NLRI updates sent/received
- o Number of Link-State NLRI updates sent/received, per neighbor
- o Number of errored received Link-State NLRI updates, per neighbor
- o Total number of locally originated Link-State NLRIs

These statistics should be recorded as absolute counts since system or session start time. An implementation MAY also enhance this information by recording peak per-second counts in each case.

7.2.6. Security Management

An operator SHOULD define an import policy to limit inbound updates as follows:

- o Drop all updates from peers that are only serving BGP-LS Consumers.

An implementation MUST have the means to limit inbound updates.

8. TLV/Sub-TLV Code Points Summary

This section contains the global table of all TLVs/sub-TLVs defined in this document.

TLV Code Point	Description	IS-IS TLV/ Sub-TLV	Reference (RFC/Section)
256	Local Node Descriptors	---	Section 4.2.1.2
257	Remote Node Descriptors	---	Section 4.2.1.3
258	Link Local/Remote Identifiers	22/4	[RFC5307] / 1.1
259	IPv4 interface address	22/6	[RFC5305] / 3.2
260	IPv4 neighbor address	22/8	[RFC5305] / 3.3
261	IPv6 interface address	22/12	[RFC6119] / 4.2
262	IPv6 neighbor address	22/13	[RFC6119] / 4.3
263	Multi-Topology ID	---	Section 4.2.2.1
264	OSPF Route Type	---	Section 4.2.3
265	IP Reachability Information	---	Section 4.2.3
512	Autonomous System	---	Section 4.2.1.4
513	BGP-LS Identifier (deprecated)	---	Section 4.2.1.4
514	OSPF Area-ID	---	Section 4.2.1.4
515	IGP Router-ID	---	Section 4.2.1.4
1024	Node Flag Bits	---	Section 4.3.1.1
1025	Opaque Node Attribute	---	Section 4.3.1.5
1026	Node Name	variable	Section 4.3.1.3
1027	IS-IS Area Identifier	variable	Section 4.3.1.2
1028	IPv4 Router-ID of Local Node	134/---	[RFC5305] / 4.3
1029	IPv6 Router-ID of Local Node	140/---	[RFC6119] / 4.1
1030	IPv4 Router-ID of Remote Node	134/---	[RFC5305] / 4.3
1031	IPv6 Router-ID of Remote Node	140/---	[RFC6119] / 4.1
1088	Administrative group (color)	22/3	[RFC5305] / 3.1

1089	Maximum link bandwidth	22/9	[RFC5305] / 3.4
1090	Max. reservable link bandwidth	22/10	[RFC5305] / 3.5
1091	Unreserved bandwidth	22/11	[RFC5305] / 3.6
1092	TE Default Metric	22/18	Section 4.3.2.3
1093	Link Protection Type	22/20	[RFC5307] / 1.2
1094	MPLS Protocol Mask	---	Section 4.3.2.2
1095	IGP Metric	---	Section 4.3.2.4
1096	Shared Risk Link Group	---	Section 4.3.2.5
1097	Opaque Link Attribute	---	Section 4.3.2.6
1098	Link Name	---	Section 4.3.2.7
1152	IGP Flags	---	Section 4.3.3.1
1153	IGP Route Tag	---	[RFC5130]
1154	IGP Extended Route Tag	---	[RFC5130]
1155	Prefix Metric	---	[RFC5305]
1156	OSPF Forwarding Address	---	[RFC2328]
1157	Opaque Prefix Attribute	---	Section 4.3.3.6

Table 12: Summary Table of TLV/Sub-TLV Code Points

9. Security Considerations

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

In the context of the BGP peerings associated with this document, a BGP speaker MUST NOT accept updates from a peer that is only providing information to a BGP-LS Consumer. That is, a participating BGP speaker should be aware of the nature of its relationships for link-state relationships and should protect itself from peers sending updates that either represent erroneous information feedback loops or are false input. Such protection can be achieved by manual configuration of consumer peers at the BGP speaker.

An operator SHOULD employ a mechanism to protect a BGP speaker against DDoS attacks from BGP-LS Consumers. The principal attack a consumer may apply is to attempt to start multiple sessions either sequentially or simultaneously. Protection can be applied by imposing rate limits.

Additionally, it may be considered that the export of link-state and TE information as described in this document constitutes a risk to confidentiality of mission-critical or commercially sensitive information about the network. BGP peerings are not automatic and require configuration; thus, it is the responsibility of the network operator to ensure that only trusted consumers are configured to receive such information.

10. Contributors

The following persons contributed significant text to RFC7752 and this document. They should be considered as co-authors.

Hannes Gredler
Rtbrick
Email: hannes@rtbrick.com

Jan Medved
Cisco Systems Inc.
USA
Email: jmedved@cisco.com

Stefano Previdi
Huawei Technologies
Italy
Email: stefano@previdi.net

Adrian Farrel
Old Dog Consulting
Email: adrian@olddog.co.uk

Saikat Ray
Individual
USA
Email: raysaikat@gmail.com

11. Acknowledgements

This document update to the BGP-LS specification [RFC7752] is a result of feedback and inputs from the discussions in the IDR working group. It also incorporates certain details and clarifications based on implementation and deployment experience with BGP-LS.

Cengiz Alaettinoglu and Parag Amritkar brought forward the need to clarify the advertisement of LAN subnet for OSPF.

We would like to thank Balaji Rajagopalan, Srihari Sangli, Shraddha Hegde, Andrew Stone, Jeff Tantsura, Acee Lindem, Jie Dong, Aijun Wang and Nandan Saha for their review and feedback on this document.

We would like to thank Robert Varga for the significant contribution he gave to RFC7752.

We would like to thank Nischal Sheth, Alia Atlas, David Ward, Derek Yeung, Murtuza Lightwala, John Scudder, Kaliraj Vairavakkalai, Les Ginsberg, Liem Nguyen, Manish Bhardwaj, Matt Miller, Mike Shand, Peter Psenak, Rex Fernando, Richard Woundy, Steven Luong, Tamas Mondal, Waqas Alam, Vipin Kumar, Naiming Shen, Carlos Pignataro, Balaji Rajagopalan, Yakov Rekhter, Alvaro Retana, Barry Leiba, and Ben Campbell for their comments on RFC7752.

12. References

12.1. Normative References

[ISO10589]
International Organization for Standardization, "Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)", ISO/IEC 10589, November 2002.

[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>.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <https://www.rfc-editor.org/info/rfc2328>.

[RFC2545] Marques, P. and F. Dupont, "Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing", RFC 2545, DOI 10.17487/RFC2545, March 1999, <https://www.rfc-editor.org/info/rfc2545>.

[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>.

[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>.

[RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, DOI 10.17487/RFC4203, October 2005, <https://www.rfc-editor.org/info/rfc4203>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC4577] Rosen, E., Psenak, P., and P. Pillay-Esnault, "OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4577, DOI 10.17487/RFC4577, June 2006, <https://www.rfc-editor.org/info/rfc4577>.

[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>.

[RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <https://www.rfc-editor.org/info/rfc4915>.

[RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <https://www.rfc-editor.org/info/rfc5036>.

[RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <https://www.rfc-editor.org/info/rfc5120>.

[RFC5130] Previdi, S., Shand, M., Ed., and C. Martin, "A Policy Control Mechanism in IS-IS Using Administrative Tags", RFC 5130, DOI 10.17487/RFC5130, February 2008, <https://www.rfc-editor.org/info/rfc5130>.

[RFC5301] McPherson, D. and N. Shen, "Dynamic Hostname Exchange Mechanism for IS-IS", RFC 5301, DOI 10.17487/RFC5301, October 2008, <https://www.rfc-editor.org/info/rfc5301>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <https://www.rfc-editor.org/info/rfc5305>.

[RFC5307] Kompella, K., Ed. and Y. Rekhter, Ed., "IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 5307, DOI 10.17487/RFC5307, October 2008, <https://www.rfc-editor.org/info/rfc5307>.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <https://www.rfc-editor.org/info/rfc5340>.

[RFC5642] Venkata, S., Harwani, S., Pignataro, C., and D. McPherson, "Dynamic Hostname Exchange Mechanism for OSPF", RFC 5642, DOI 10.17487/RFC5642, August 2009, <https://www.rfc-editor.org/info/rfc5642>.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, <https://www.rfc-editor.org/info/rfc5890>.

[RFC6119] Harrison, J., Berger, J., and M. Bartlett, "IPv6 Traffic Engineering in IS-IS", RFC 6119, DOI 10.17487/RFC6119, February 2011, <https://www.rfc-editor.org/info/rfc6119>.

[RFC6549] Lindem, A., Roy, A., and S. Mirtorabi, "OSPFv2 Multi-Instance Extensions", RFC 6549, DOI 10.17487/RFC6549, March 2012, <https://www.rfc-editor.org/info/rfc6549>.

[RFC6565] Pillay-Esnault, P., Moyer, P., Doyle, J., Ertekin, E., and M. Lundberg, "OSPFv3 as a Provider Edge to Customer Edge (PE-CE) Routing Protocol", RFC 6565, DOI 10.17487/RFC6565, June 2012, <https://www.rfc-editor.org/info/rfc6565>.

[RFC7606] Chen, E., Ed., Scudder, J., Ed., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP UPDATE Messages", RFC 7606, DOI 10.17487/RFC7606, August 2015, <https://www.rfc-editor.org/info/rfc7606>.

[RFC7684] Psenak, P., Gredler, H., Shakir, R., Henderickx, W., Tantsura, J., and A. Lindem, "OSPFv2 Prefix/Link Attribute Advertisement", RFC 7684, DOI 10.17487/RFC7684, November 2015, <https://www.rfc-editor.org/info/rfc7684>.

[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>.

[RFC8202] Ginsberg, L., Previdi, S., and W. Henderickx, "IS-IS Multi-Instance", RFC 8202, DOI 10.17487/RFC8202, June 2017, <https://www.rfc-editor.org/info/rfc8202>.

[RFC8362] Lindem, A., Roy, A., Goethals, D., Reddy Vallem, V., and F. Baker, "OSPFv3 Link State Advertisement (LSA) Extensibility", RFC 8362, DOI 10.17487/RFC8362, April 2018, <https://www.rfc-editor.org/info/rfc8362>.

[RFC8654] Bush, R., Patel, K., and D. Ward, "Extended Message Support for BGP", RFC 8654, DOI 10.17487/RFC8654, October 2019, <https://www.rfc-editor.org/info/rfc8654>.

12.2. Informative References

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <https://www.rfc-editor.org/info/rfc1918>.

[RFC4272] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272, DOI 10.17487/RFC4272, January 2006, <https://www.rfc-editor.org/info/rfc4272>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[RFC4655] Farrel, A., Vasseur, J., 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>.

[RFC5152] Vasseur, JP., Ed., Ayyangar, A., Ed., and R. Zhang, "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)", RFC 5152, DOI 10.17487/RFC5152, February 2008, <https://www.rfc-editor.org/info/rfc5152>.

[RFC5316] Chen, M., Zhang, R., and X. Duan, "ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5316, DOI 10.17487/RFC5316, December 2008, <https://www.rfc-editor.org/info/rfc5316>.

[RFC5392] Chen, M., Zhang, R., and X. Duan, "OSPF Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5392, DOI 10.17487/RFC5392, January 2009, <https://www.rfc-editor.org/info/rfc5392>.

[RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, DOI 10.17487/RFC5693, October 2009, <https://www.rfc-editor.org/info/rfc5693>.

[RFC5706] Harrington, D., "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions", RFC 5706, DOI 10.17487/RFC5706, November 2009, <https://www.rfc-editor.org/info/rfc5706>.

[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>.

[RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014, <https://www.rfc-editor.org/info/rfc7285>.

[RFC7770] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, DOI 10.17487/RFC7770, February 2016, <https://www.rfc-editor.org/info/rfc7770>.

[RFC7911] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", RFC 7911, DOI 10.17487/RFC7911, July 2016, <https://www.rfc-editor.org/info/rfc7911>.

Appendix A. Changes from RFC 7752

This section lists the high-level changes from RFC 7752 and provides reference to the document sections wherein those have been introduced.

1.   Update the Figure 1 in Section 1 and added Section 3 to illustrate the different roles of a BGP implementation in conveying link-state information.

2.   In Section 4.1, clarification about the TLV handling aspects that are applicable to both the NLRI and BGP-LS Attribute parts and those that are applicable only for the NLRI portion. An implementation may have missed the part about handling of unrecognized TLV and so, based on [RFC7606] guidelines, might discard the unknown NLRI types. This aspect is now unambiguously clarified in Section 4.2. Also, the TLVs in the BGP-LS Attribute that are not ordered are not to be considered as malformed.

3.   Clarification of mandatory and optional TLVs in both NLRI and BGP-LS Attribute portions all through the document.

4.   Handling of large size of BGP-LS Attribute with growth in BGP-LS information is explained in Section 4.3 along with mitigation of errors arising out of it.

5.   Clarified that the document describes the NLRI descriptor TLVs for the protocols and NLRI types specified in this document and future BGP-LS extensions must describe the same for other protocols and NLRI types that they introduce.

6.   Clarification on the use of Identifier field in the Link-State NLRI in Section 4.2 is provided. It was defined ambiguously to refer to only mutli-instance IGP on a single link while it can also be used for multiple IGP protocol instances on a router. The IANA registry is accordingly being removed.

7.   The BGP-LS Identifier TLV in the Node Descriptors has been deprecated. Its use was not well specified by [RFC7752] and there has been some amount of confusion between implementators on its usage for identification of IGP domains as against the use of the Identifier doing the same functionality as the Instance-ID when running multiple instances of IGP routing protocols.

8.   Clarification that the Area-ID TLV is mandatory in the Node Descriptor for origination of information from OSPF except for

when sourcing information from AS-scope LSAs where this TLV is not applicable.

9. Moved MT-ID TLV from the Node Descriptor section to under the Link Descriptor section since it is not a Node Descriptor sub-TLV. Fixed the ambiguity in the encoding of OSPF MT-ID in this TLV. Updated the IS-IS specification reference section and describe the differences in the applicability of the R flags when MT-ID TLV is used as link descriptor TLV and Prefix Attribute TLV. MT-ID TLV use is now elevated to SHOULD when it is enabled in the underlying IGP.

10. Clarified that IPv6 Link-Local Addresses are not advertised in the Link Descriptor TLVs and the local/remote identifiers are to be used instead for links with IPv6 link-local addresses only.

11. Update the usage of OSPF Route Type TLV to mandate its use for OSPF prefixes in Section 4.2.3.1 since this is required for segregation of intra-area prefixes that are used to reach a node (e.g. a loopback) from other types of inter-area and external prefixes.

12. Clarification on the specific OSPFv2 and OSPFv3 protocol TLV space to be used in the node, link and prefix opaque attribute TLVs.

13. Clarification on the length of the Node Flag Bits and IGP Flags TLVs to be one octet.

14. Updated the Node Name TLV in Section 4.3.1.3 with the OSPF specification.

15. Clarification on the size of the IS-IS Narrow Metric advertisement via the IGP Metric TLV and the handling of the unused bits.

16. Clarified the advertisement of the prefix corresponding to the LAN segment in an OSPF network in Section 4.11.

17. Clarified the advertisement and support for OSPF specific concepts like Virtual links, Sham links and Type 4 LSAs in Section 4.7 and Section 4.8.

18. Introduced Private Use TLV code point space and specified their encoding in Section 4.4.

19. Introduced Section 4.9 where issues related to consistency of reporting IGP link-state along with their solutions are covered.

20. Added recommendation for isolation of BGP-LS sessions from other BGP route exchange to avoid errors and faults in BGP-LS affecting the normal BGP routing.

21. Updated the Fault Management section with detailed rules based on the role in the BGP-LS information propagation flow.

22. Change to the management of BGP-LS IANA registries from "Specification Required" to "Expert Review" along with updated guidelines for Designated Experts.

23. Added BGP-LS IANA registries with "Specification Required" policy for the flag fields of various TLVs that was missed out.

Author's Address

Ketan Talaulikar (editor)
Cisco Systems
India

Email: ketant@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 27, 2021

Z. Li
Huawei
L. Ou
Y. Luo
China Telcom Co., Ltd.
S. Lu
Tencent
G. Mishra
Verizon Inc.
H. Chen
Futurewei
S. Zhuang
H. Wang
Huawei

November 23, 2020

BGP Extensions for Routing Policy Distribution (RPD)
draft-ietf-idr-rpd-10

Abstract

It is hard to adjust traffic and optimize traffic paths in a traditional IP network from time to time through manual configurations. It is desirable to have a mechanism for setting up routing policies, which adjusts traffic and optimizes traffic paths automatically. This document describes BGP Extensions for Routing Policy Distribution (BGP RPD) to support this.

Requirements Language

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

It is difficult to optimize traffic paths in a traditional IP network because of the following:

- Complex and error prone configuration. Traffic can only be adjusted device by device. The configurations on all the routers that the traffic traverses need to be changed or added. There are already lots of policies configured on the routers in an operational network. There are different types of policies, which include security, management and control policies. These policies are relatively stable. However, the policies for adjusting traffic are dynamic. Whenever the traffic through a route is not expected, the policies to adjust the traffic for that route are configured on the related routers. It is complex and error prone to dynamically add or change the policies to the existing policies on the special routers to adjust the traffic.

- Difficult maintenance. The routing policies used to control network routes are dynamic, posing difficulties to subsequent maintenance. High maintenance skills are required.

It is desirable to have an automatic mechanism for setting up routing policies, which can simplify routing policy configuration. This document describes extensions to BGP for Routing Policy Distribution to resolve these issues.

2. Terminology

The following terminology is used in this document.

- ACL: Access Control List

- BGP: Border Gateway Protocol [RFC4271]

- FS: Flow Specification

- NLRI: Network Layer Reachability Information [RFC4271]

- PBR: Policy-Based Routing

- RPD: Routing Policy Distribution

- VPN: Virtual Private Network

3. Problem Statement

Providers have the requirement to adjust their business traffic routing policies from time to time because of the following:

- Business development or network failure introduces link congestion and overload.

- Business changes or network additions produce unused resources such as idle links.

- Network transmission quality is decreased as the result of delay, loss and they need to adjust traffic to other paths.

- To control OPEX and CPEX, they may prefer the transit provider with lower price.

3.1. Inbound Traffic Control

In Figure 1, for the reasons above, the provider P of AS100 may wish the inbound traffic from AS200 to enter AS100 through link L3 instead of the others. Since P doesn’t have any administrative control over AS200, there is no way for P to directly modify the route selection criteria inside AS200.

```
                  Traffic from PE1 to Prefix1
             ----------------------------------->

   +-----------------+            +-------------------------+
   |     +---------+ |        L1  | +----+      +----------+|
   |     |Speaker1 | +------------+ |IGW1|      |policy    ||
   |     +---------+ |**      L2**| +----+      |controller||
   |                 |  **    **  |             +----------+|
   | +---+           |    ****    |                         |
   | |PE1|           |    ****    |                         |
   | +---+           |  **    **  |                         |
   |     +---------+ |**      L3**| +----+                  |
   |     |Speaker2 | +------------+ |IGW2|      AS100       |
   |     +---------+ |        L4  | +----+                  |
   |                 |            |                         |
   |    AS200        |            |                         |
   |                 |            |  ...                    |
   |     +---------+ |            | +----+      +-------+   |
   |     |Speakern | |            | |IGWn|      |Prefix1|   |
   |     +---------+ |            | +----+      +-------+   |
   +-----------------+            +-------------------------+

               Prefix1 advertised from AS100 to AS200
             <----------------------------------------
```

Figure 1: Inbound Traffic Control case

3.2. Outbound Traffic Control

In Figure 2, the provider P of AS100 prefers link L3 for the traffic to the destination Prefix2 among multiple exits and links to AS200. This preference can be dynamic and might change frequently because of the reasons above. So, provider P expects an efficient and convenient solution.

```
                  Traffic from PE2 to Prefix2
             ----------------------------------->
   +-------------------------+            +------------------+
   |+----------+      +----+ |L1          | +---------+      |
   ||policy    |      |IGW1| +------------+ |Speaker1 |      |
   ||controller|      +----+ |**        **| +---------+      |
   |+----------+             |L2**    **  |        +-------+|
   |                         |    ****    |        |Prefix2||
   |                         |    ****    |        +-------+|
   |                         |L3**    **  |                  |
   |      AS100       +----+ |**        **| +---------+      |
   |                  |IGW2| +------------+ |Speaker2 |      |
   |                  +----+ |L4          | +---------+      |
   |                         |            |                  |
   |+---+                    |            |    AS200         |
   ||PE2|              ...   |            |                  |
   |+---+                    |            |                  |
   |                  +----+ |            | +---------+      |
   |                  |IGWn| |            | |Speakern |      |
   |                  +----+ |            | +---------+      |
   +-------------------------+            +------------------+

               Prefix2 advertised from AS200 to AS100
             <----------------------------------------
```

Figure 2: Outbound Traffic Control case

4. Protocol Extensions

This document specifies a solution using a new AFI and SAFI with the BGP Wide Community for encoding a routing policy.

4.1. Using a New AFI and SAFI

A new AFI and SAFI are defined: the Routing Policy AFI whose codepoint 16398 has been assigned by IANA, and SAFI whose codepoint 75 has been assigned by IANA.

The AFI and SAFI pair uses a new NLRI, which is defined 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 Length  |
   +-+-+-+-+-+-+-+-+
   |  Policy Type  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Distinguisher (4 octets)                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Peer IP (4/16 octets)                    ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

NLRI Length: 1 octet represents the length of NLRI. If the Length is anything other than 9 or 21, the NLRI is corrupt and the enclosing UPDATE message MUST be ignored.

Policy Type: 1 octet indicates the type of a policy. 1 is for Export policy. 2 is for Import policy. If the Policy Type is any other value, the NLRI is corrupt and the enclosing UPDATE message MUST be ignored.

Distinguisher: 4 octet value uniquely identifies the content/ policy. It is used to sort/order the polices from the lower to higher distinguisher. They are applied in order. The policy with a lower/smaller distinguisher is applied before the policies with higher/larger distinguishers.

Peer IP: 4/16 octet value indicates IPv4/IPv6 peers. Its default value is 0, which indicates that when receiving a BGP UPDATE message with the NLRI, a BGP speaker will apply the policy in the message to all its IPv4/IPv6 peers.

Under RPD AFI/SAFI, the RPD routes are stored and ordered according to their keys. Under IPv4/IPv6 Unicast AFI/SAFI, there are IPv4/IPv6 unicast routes learned and various static policies configured. In addition, there are dynamic RPD policies from the RPD AFI/SAFI when RPD is enabled.

Before advertising an IPv4/IPv6 Unicast AFI/SAFI route, the configured policies are applied to it first, and then the RPD Export policies are applied.

The NLRI containing the Routing Policy is carried in MP_Reach_NLRI and MP_UNREACH_NLRI path attributes in a BGP UPDATE message, which MUST also contain the BGP mandatory attributes and MAY contain some BGP optional attributes.

When receiving a BGP UPDATE message with routing policy, a BGP speaker processes it as follows:

- o If the peer IP in the NLRI is 0, then apply the routing policy to all the remote peers of this BGP speaker.

- o If the peer IP in the NLRI is non-zero, then the IP address indicates a remote peer of this BGP speaker and the routing policy will be applied to it.

The content of the Routing Policy is encoded in a BGP Wide Community.

4.2. BGP Wide Community and Atoms

The BGP wide community is defined in [I-D.ietf-idr-wide-bgp-communities]. It can be used to facilitate the delivery of new network services and be extended easily for distributing different kinds of routing policies.

A wide community Atom is a TLV (or sub-TLV), which may be included in a BGP wide community container (or BGP wide community for short) containing some BGP Wide Community TLVs. Three BGP Wide Community TLVs are defined in [I-D.ietf-idr-wide-bgp-communities], which are BGP Wide Community Target(s) TLV, Exclude Target(s) TLV, and Parameter(s) TLV. The value of each of these TLVs comprises a series of Atoms, each of which is a TLV (or sub-TLV). A new wide community Atom is defined for BGP Wide Community Target(s) TLV and a few new Atoms are defined for BGP Wide Community Parameter(s) TLV. For your reference, the format of the TLV is illustrated below:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-++-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |             Length             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         Value (variable)                      ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Format of Wide Community Atom TLV

4.2.1. RouteAttr TLV/sub-TLV

A RouteAttr Atom TLV (or RouteAttr TLV/sub-TLV for short) is defined and may be included in a Target TLV. 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 (TBD1)  |        Length (variable)         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                           sub-TLVs                            ~
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Format of RouteAttr Atom TLV

The Type for RouteAttr is TBD1. In RouteAttr TLV, four sub-TLVs are defined: IPv4 Prefix, IPv6 Prefix, AS-Path, and Community sub-TLV.

An IP prefix sub-TLV gives matching criteria on IPv4 prefixes. Its format is illustrated below:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Type  1      |          Length (N x 8)        |M-Type | Flags |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                           IPv4 Address                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Mask      |    GeMask     |     LeMask     |M-Type | Flags |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ~        . . .
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                           IPv4 Address                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Mask      |    GeMask     |     LeMask     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Format of IPv4 Prefix sub-TLV

Type: 1 for IPv4 Prefix.

Length: N x 8, where N is the number of tuples <M-Type, Flags, IPv4 Address, Mask, GeMask, LeMask>. If Length is not a multiple of 8, the Atom is corrupt and the enclosing UPDATE message MUST be ignored.

M-Type: 4 bits for match types, four of which are defined:

M-Type = 0: Exact match.

M-Type = 1: Match prefix greater and equal to the given masks.

M-Type = 2: Match prefix less and equal to the given masks.

M-Type = 3: Match prefix within the range of the given masks.

Flags: 4 bits. No flags are currently defined.

IPv4 Address: 4 octets for an IPv4 address.

Mask: 1 octet for the mask length.

GeMask: 1 octet for match range, must be less than Mask or be 0.

LeMask: 1 octet for match range, must be greater than Mask or be 0.

For example, tuple <M-Type=0, Flags=0, IPv4 Address = 1.1.0.0, Mask = 22, GeMask = 0, LeMask = 0> represents an exact IP prefix match for 1.1.0.0/22.

<M-Type=1, Flags=0, IPv4 Address = 16.1.0.0, Mask = 24, GeMask = 24, LeMask = 0> represents match IP prefix 1.1.0.0/24 greater-equal 24.

<M-Type=2, Flags=0, IPv4 Address = 17.1.0.0, Mask = 24, GeMask = 0, LeMask = 26> represents match IP prefix 17.1.0.0/24 less-equal 26.

<M-Type=3, Flags=0, IPv4 Address = 18.1.0.0, Mask = 24, GeMask = 24, LeMask = 32> represents match IP prefix 18.1.0.0/24 greater-equal to 24 and less-equal 32.

Similarly, an IPv6 Prefix sub-TLV represents match criteria on IPv6 prefixes. Its format is illustrated below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type  4     |         Length (N x 20)       |M-Type | Flags |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     IPv6 Address (16 octets)                  ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Mask      |     GeMask    |     LeMask    |M-Type | Flags |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~       . . .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      IPv6 Address (16 octets                  ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Mask      |     GeMask    |     LeMask    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Format of IPv6 Prefix sub-TLV

An AS-Path sub-TLV represents a match criteria in a regular expression string. Its format is illustrated below:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Type  2      |      Length (Variable)        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    AS-Path Regex String                       |
   :                                                               :
   |                                                               ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Format of AS Path sub-TLV

Type: 2 for AS-Path.

Length: Variable, maximum is 1024.

AS-Path Regex String: AS-Path regular expression string.

A community sub-TLV represents a list of communities to be matched all. Its format is illustrated below:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Type  3      |        Length (N x 4 + 1)       |    Flags    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Community 1 Value                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                              . . .                            ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Community N Value                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Format of Community sub-TLV

Type: 3 for Community.

Length: N x 4 + 1, where N is the number of communities. If Length is not a multiple of 4 plus 1, the Atom is corrupt and the enclosing UPDATE MUST be ignored.

Flags: 1 octet. No flags are currently defined. These bits MUST be sent as zero and ignored on receipt.

4.2.2. Sub-TLVs of the Parameters TLV

This document introduces 2 community values:

MATCH AND SET ATTR: If the IPv4/IPv6 unicast routes to a remote peer match the specific conditions defined in the routing policy extracted from the RPD route, then the attributes of the IPv4/IPv6 unicast routes will be modified when sending to the remote peer per the actions defined in the RPD route.

MATCH AND NOT ADVERTISE: If the IPv4/IPv6 unicast routes to a remote peer match the specific conditions defined in the routing policy extracted from the RPD route, then the IPv4/IPv6 unicast routes will not be advertised to the remote peer.

For the Parameter(s) TLV, two action sub-TLVs are defined: MED change sub-TLV and AS-Path change sub-TLV. When the community in the container is MATCH AND SET ATTR, the Parameter(s) TLV can include these sub-TLVs. When the community is MATCH AND NOT ADVERTISE, the Parameter(s) TLV's value is empty.

A MED change sub-TLV indicates an action to change the MED. Its format is illustrated below:

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

Format of MED Change sub-TLV

Type: 1 for MED Change.

Length: 5. If Length is any other value, the sub-TLV is corrupt and the enclosing UPDATE MUST be ignored.

OP: 1 octet. Three are defined:

OP = 0: assign the Value to the existing MED.

OP = 1: add the Value to the existing MED. If the sum is greater than the maximum value for MED, assign the maximum value to MED.

OP = 2: subtract the Value from the existing MED. If the existing MED minus the Value is less than 0, assign 0 to MED.

If OP is any other value, the sub-TLV is ignored.

Value: 4 octets.

An AS-Path change sub-TLV indicates an action to change the AS-Path. Its format is illustrated below:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Type  2      |        Length (n x 5)         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                             AS1                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    Count1     |
   +-+-+-+-+-+-+-+-+
   ~       . . .
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                             ASn                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    Countn     |
   +-+-+-+-+-+-+-+-+
```

Format of AS-Path Change sub-TLV

Type: 2 for AS-Path Change.

Length: n x 5. If Length is not a multiple of 5, the sub-TLV is corrupt and the enclosing UPDATE MUST be ignored.

ASi: 4 octet. An AS number.

Counti: 1 octet. ASi repeats Counti times.

The sequence of AS numbers are added to the existing AS Path.

4.3. Capability Negotiation

It is necessary to negotiate the capability to support BGP Extensions for Routing Policy Distribution (RPD). The BGP RPD Capability is a new BGP capability [RFC5492]. The Capability Code for this capability is 72 assigned by the IANA. The Capability Length field of this capability is variable. The Capability Value field consists of one or more of the following tuples:

```
+--------------------------------------------------+
|  Address Family Identifier (2 octets)            |
+--------------------------------------------------+
|  Subsequent Address Family Identifier (1 octet)  |
+--------------------------------------------------+
|  Send/Receive (1 octet)                          |
+--------------------------------------------------+
```

BGP RPD Capability

The meaning and use of the fields are as follows:

Address Family Identifier (AFI): This field is the same as the one used in [RFC4760].

Subsequent Address Family Identifier (SAFI): This field is the same as the one used in [RFC4760].

Send/Receive: This field indicates whether the sender is (a) willing to receive Routing Policies from its peer (value 1), (b) would like to send Routing Policies to its peer (value 2), or (c) both (value 3) for the <AFI, SAFI>. If Send/Receive is any other value, that tuple is ignored but any other tuples present are still used.

5. Operations

This section presents a typical application scenario and some details about handling a related failure.

5.1. Application Scenario

Figure 3 illustrates a typical scenario, where RPD is used by a controller with a Route Reflector (RR) to adjust traffic dynamically.

```
            +--------------+
            |  Controller  |
            +-------+------+
                     \
                      \ RPD
                    .--\._.+--+                       ___...__
                __(     \       '.---...             (         )
               /      RR o -------- A o) ---------- (o X   AS2  )
              (o E       |\             )     _____//(___   ___)
               (         | \_______ B o) ____/     /     '''
                (o F      \           )       ____/
                 (         \_____ C o) ______/         ___...__
                  '    AS1        _)  \_____          (         )
                   '---._.-.     )          \_______ (o Y   AS3  )
                            '___'                     (___   ___)
                                                          '''
```

Figure 3: Controller with RR Adjusts Traffic

The controller connects the RR through a BGP session. There is a BGP session between the RR and each of routers A, B and C in AS1, which is shown in the figure. Other sessions in AS1 are not shown in the figure.

There is router X in AS2. There is a BGP session between X and each of routers A, B and C in AS1.

There is router Y in AS3. There is a BGP session between Y and router C in AS1.

The controller sends a RPD route to the RR. After receiving the RPD route from the controller, the RR reflects the RPD route to routers A, B and C. After receiving the RPD route from the RR, routers A, B and C extract the routing policy from the RPD route. If the peer IP in the NLRI of the RPD route is 0, then apply the routing policy to all the remote peers of routers A, B and C. If the peer IP in the NLRI of the RPD route is non-zero, then the IP address indicates a remote peer of routers A, B and C and such routing policy is applied to the specific remote peer. The IPv4/IPv6 unicast routes towards router X in AS2 and router Y in AS3 will be adjusted based on the routing policy sent by the controller via a RPD route.

The controller uses the RT extend community to notify a router whether to receive a RPD policy. For example, if there is not any adjustment on router B, the controller sends RPD routes with the RTs for A and C. B will not receive the routes.

The process of adjusting traffic in a network is a close loop. The loop starts from the controller with some traffic expectations on a set of routes. The controller obtains the information about traffic flows for the related routes. It analyzes the traffic and checks whether the current traffic flows meet the expectations. If the expectations are not met, the controller adjusts the traffic. And then the loop goes to the starter of the loop (The controller obtains the information about traffic ...).

5.2. About Failure

A RPD route is not a configuration. When it is sent to a router, no ack is needed from the router. The existing BGP mechanisms are re-used for delivering a RPD route. After the route is delivered to a router, it will be successful. This is guaranteed by the BGP protocols.

If there is a failure for the router to install the route locally, this failure is a bug of the router. The bug needs to be fixed.

For the errors mentioned in [RFC7606], they are handled according to [RFC7606]. These errors are bugs, which need to be resolved.

Regarding to the failure of the controller, some existing mechanisms such BGP GR [RFC4724] and BGP Long-lived Graceful Restart (LLGR) can be used to let the router keep the routes from the controller for some time.

With support of "Long-lived Graceful Restart Capability" [I-D.ietf-idr-long-lived-gr], the routes can be retained for a longer time after the controller fails.

In the worst case, the controller fails and the RPD routes for adjusting the traffic are withdrawn. The traffic adjusted/redirected may take its old path. This should be acceptable.

6. Contributors

The following people have substantially contributed to the definition of the BGP-FS RPD and to the editing of this document:

Peng Zhou
Huawei
Email: Jewpon.zhou@huawei.com

7. Security Considerations

Protocol extensions defined in this document do not affect BGP security other than as discussed in the Security Considerations section of [RFC5575].

8. Acknowledgements

The authors would like to thank Acee Lindem, Jeff Haas, Jie Dong, Lucy Yong, Qiandeng Liang, Zhenqiang Li, Robert Raszuk, Donald Eastlake, Ketan Talaulikar, and Jakob Heitz for their comments to this work.

9. IANA Considerations

9.1. Existing Assignments

IANA has assigned a new AFI of value 16398 from the registry "Address Family Numbers" for Routing Policy.

IANA has assigned a new SAFI of value 75 from the registry "Subsequent Address Family Identifiers (SAFI) Parameters" for Routing Policy.

IANA has assigned a new Code Point of value 72 from the registry "Capability Codes" for Routing Policy Distribution.

9.2. Routing Policy Type Registry

IANA is requested to create a new registry called "Routing Policy Type". The allocation policy of this registry is "First Come First Served (FCFS)".

The initial code points are as follows:

Code Point	Description	Reference
0	Reserved	
1	Export Policy	This document
2	Import Policy	This document
3 - 255	Available	

9.3. RouteAttr Atom Type

IANA is requested to assign a code-point from the registry "BGP Community Container Atom Types" as follows:

TLV Code Point	Description	Reference
TBD1 (48 suggested)	RouteAttr Atom	This document

9.4. Route Attributes Sub-TLV Registry

IANA is requested to create a new registry called "Route Attributes Sub-TLV" under RouteAttr Atom TLV. The allocation policy of this registry is "First Come First Served (FCFS)".

The initial code points are as follows:

Code Point	Description	Reference
0	Reserved	
1	IPv4 Prefix Sub-TLV	This document
2	AS-Path Sub-TLV	This document
3	Community Sub-TLV	This document
4	IPv6 Prefix Sub-TLV	This document
5 - 255	Available	

9.5. Attribute Change Sub-TLV Registry

IANA is requested to create a new registry called "Attribute Change Sub-TLV" under Parameter(s) TLV. The allocation policy of this registry is "First Come First Served (FCFS)".

Initial code points are as follows:

Code Point	Description	Reference
0	Reserved	
1	MED Change Sub-TLV	This document
2	AS-Path Change Sub-TLV	This document
3 - 255	Available	

10. References

10.1. Normative References

[I-D.ietf-idr-wide-bgp-communities]
Raszuk, R., Haas, J., Lange, A., Decraene, B., Amante, S., and P. Jakma, "BGP Community Container Attribute", draft-ietf-idr-wide-bgp-communities-05 (work in progress), July 2018.

[RFC1997] Chandra, R., Traina, P., and T. Li, "BGP Communities Attribute", RFC 1997, DOI 10.17487/RFC1997, August 1996, <https://www.rfc-editor.org/info/rfc1997>.

[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>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[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>.

[RFC5492] Scudder, J. and R. Chandra, "Capabilities Advertisement with BGP-4", RFC 5492, DOI 10.17487/RFC5492, February 2009, <https://www.rfc-editor.org/info/rfc5492>.

[RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <https://www.rfc-editor.org/info/rfc5575>.

[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>.

10.2. Informative References

[I-D.ietf-idr-long-lived-gr] Uttaro, J., Chen, E., Decraene, B., and J. Scudder, "Support for Long-lived BGP Graceful Restart", draft-ietf-idr-long-lived-gr-00 (work in progress), September 2019.

[I-D.ietf-idr-registered-wide-bgp-communities] Raszuk, R. and J. Haas, "Registered Wide BGP Community Values", draft-ietf-idr-registered-wide-bgp-communities-02 (work in progress), May 2016.

[RFC4724] Sangli, S., Chen, E., Fernando, R., Scudder, J., and Y. Rekhter, "Graceful Restart Mechanism for BGP", RFC 4724, DOI 10.17487/RFC4724, January 2007, <https://www.rfc-editor.org/info/rfc4724>.

[RFC7606] Chen, E., Ed., Scudder, J., Ed., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP UPDATE Messages", RFC 7606, DOI 10.17487/RFC7606, August 2015, <https://www.rfc-editor.org/info/rfc7606>.

Authors' Addresses

Zhenbin Li
Huawei
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: lizhenbin@huawei.com

Liang Ou
China Telcom Co., Ltd.
109 West Zhongshan Ave,Tianhe District
Guangzhou 510630
China

Email: ouliang@chinatelecom.cn

Yujia Luo
China Telcom Co., Ltd.
109 West Zhongshan Ave,Tianhe District
Guangzhou 510630
China

Email: luoyuj@sdu.edu.cn

Sujian Lu
Tencent
Tengyun Building,Tower A ,No. 397 Tianlin Road
Shanghai, Xuhui District 200233
China

Email: jasonlu@tencent.com

Gyan S. Mishra
Verizon Inc.
13101 Columbia Pike
Silver Spring MD 20904
USA

Phone: 301 502-1347
Email: gyan.s.mishra@verizon.com

Huaimo Chen
Futurewei
Boston, MA
USA

Email: Huaimo.chen@futurewei.com

Shunwan Zhuang
Huawei
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: zhuangshunwan@huawei.com

Haibo Wang
Huawei
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: rainsword.wang@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 17, 2021

S. Previdi
Individual
C. Filsfils
K. Talaulikar, Ed.
Cisco Systems
P. Mattes
Microsoft
E. Rosen
Juniper Networks
D. Jain
S. Lin
Google

November 13, 2020

Advertising Segment Routing Policies in BGP
draft-ietf-idr-segment-routing-te-policy-11

Abstract

This document defines a new BGP SAFI with a new NLRI in order to advertise a candidate path of a Segment Routing (SR) Policy. An SR Policy is a set of candidate paths, each consisting of one or more segment lists. The headend of an SR Policy may learn multiple candidate paths for an SR Policy. Candidate paths may be learned via a number of different mechanisms, e.g., CLI, NetConf, PCEP, or BGP. This document specifies the way in which BGP may be used to distribute SR Policy candidate paths. New sub-TLVs for the Tunnel Encapsulation Attribute are defined for signaling information about these candidate paths.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Segment Routing (SR) [RFC8402] allows a headend node to steer a packet flow along any path. Intermediate per-flow states are eliminated thanks to source routing.

The headend node is said to steer a flow into a SR Policy.

The header of a packet steered in an SR Policy is augmented with the ordered list of segments associated with that SR Policy.

[I-D.ietf-spring-segment-routing-policy] details the concepts of SR Policy and steering into an SR Policy. These apply equally to the MPLS and IPv6 (known as SRv6) data plane instantiations of Segment Routing with their respective representations of segments as SR-MPLS SID and SRv6 SID as described in [RFC8402].

The SR Policy related functionality described in [I-D.ietf-spring-segment-routing-policy] can be conceptually viewed as being incorporated in an SR Policy Module (SRPM). Following is a reminder of the high-level functionality of SRPM:

- Learning multiple candidate paths for an SR Policy via various mechanisms (CLI, NetConf, PCEP or BGP).

- Selection of the best candidate path for an SR Policy.

- Binding BSID to the selected candidate path of an SR Policy.

- Installation of the selected candidate path and its BSID in the forwarding plane.

This document specifies the way to use BGP to distribute one or more of the candidate paths of an SR Policy to the headend of that policy. The document describes the functionality that provided by BGP and, as appropriate, provides references for the functionality which is outside the scope of BGP (i.e. resides within SRPM on the headend node).

This document specifies a way of representing SR Policy candidate paths in BGP UPDATE messages. BGP can then be used to propagate the SR Policy candidate paths to the headend nodes in the network. The usual BGP rules for BGP propagation and best-path selection are used. At the headend of a specific policy, this will result in one or more candidate paths being installed into the "BGP table". These paths are then passed to the SRPM. The SRPM may compare them to candidate paths learned via other mechanisms, and will choose one or more paths to be installed in the data plane. BGP itself does not install SR Policy candidate paths into the data plane.

This document defines a new BGP address family (SAFI). In UPDATE messages of that address family, the NLRI identifies an SR Policy Candidate Path, and the attributes encode the segment lists and other details of that SR Policy Candidate Path.

While for simplicity we may write that BGP advertises an SR Policy, it has to be understood that BGP advertises a candidate path of an SR policy and that this SR Policy might have several other candidate paths provided via BGP (via an NLRI with a different distinguisher as defined in this document), PCEP, NETCONF or local policy configuration.

Typically, a controller defines the set of policies and advertise them to policy head-end routers (typically ingress routers). The policy advertisement uses BGP extensions defined in this document. The policy advertisement is, in most but not all of the cases, tailored for a specific policy head-end. In this case the advertisement may be sent on a BGP session to that head-end and not propagated any further.

Alternatively, a router (i.e., a BGP egress router) advertises SR Policies representing paths to itself. In this case, it is possible to send the policy to each head-end over a BGP session to that head-end, without requiring any further propagation of the policy.

An SR Policy intended only for the receiver will, in most cases, not traverse any Route Reflector (RR, [RFC4456]).

In some situations, it is undesirable for a controller or BGP egress router to have a BGP session to each policy head-end. In these situations, BGP Route Reflectors may be used to propagate the advertisements, or it may be necessary for the advertisement to propagate through a sequence of one or more AS. To make this possible, an attribute needs to be attached to the advertisement that enables a BGP speaker to determine whether it is intended to be a head-end for the advertised policy. This is done by attaching one or more Route Target Extended Communities to the advertisement ([RFC4360]).

The BGP extensions for the advertisement of SR Policies include following components:

- A new Subsequent Address Family Identifier (SAFI) whose NLRI identifies an SR Policy candidate path.

- A new Tunnel Type identifier for SR Policy, and a set of sub-TLVs to be inserted into the Tunnel Encapsulation Attribute (as defined in [I-D.ietf-idr-tunnel-encaps]) specifying segment lists of the SR Policy candidate path, as well as other information about the SR Policy.

- One or more IPv4 address format route-target extended community ([RFC4360]) attached to the SR Policy advertisement and that indicates the intended head-end of such SR Policy advertisement.

- The Color Extended Community (as defined in [I-D.ietf-idr-tunnel-encaps]) and used in order to steer traffic into an SR Policy, as described in section 8.4 in [I-D.ietf-spring-segment-routing-policy]. This document (Section 3) modifies the format of the Color Extended Community by using the two leftmost bits of the RESERVED field.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "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. SR Policy Encoding

2.1. SR Policy SAFI and NLRI

A new SAFI is defined: the SR Policy SAFI with codepoint 73. The AFI used MUST be IPv4(1) or IPv6(2).

The SR Policy SAFI uses a new NLRI defined as follows:

```
+------------------+
|  NLRI Length     | 1 octet
+------------------+
|  Distinguisher   | 4 octets
+------------------+
|  Policy Color    | 4 octets
+------------------+
|  Endpoint        | 4 or 16 octets
+------------------+
```

where:

- o NLRI Length: 1 octet of length expressed in bits as defined in [RFC4760]. When AFI = 1 value MUST be 96 and when AFI = 2 value MUST be 192.

- o Distinguisher: 4-octet value uniquely identifying the policy in the context of <color, endpoint> tuple. The distinguisher has no semantic value and is solely used by the SR Policy originator to make unique (from an NLRI perspective) multiple candidate paths of the same SR Policy.

- o Policy Color: 4-octet value identifying (with the endpoint) the policy. The color is used to match the color of the destination prefixes to steer traffic into the SR Policy as specified in [I-D.ietf-spring-segment-routing-policy].

- o Endpoint: identifies the endpoint of a policy. The Endpoint may represent a single node or a set of nodes (e.g., an anycast address). The Endpoint is an IPv4 (4-octet) address or an IPv6 (16-octet) address according to the AFI of the NLRI.

The color and endpoint are used to automate the steering of BGP Payload prefixes on SR Policy as described in [I-D.ietf-spring-segment-routing-policy].

The NLRI containing the SR Policy candidate path is carried in a BGP UPDATE message [RFC4271] using BGP multi-protocol extensions [RFC4760] with an AFI of 1 or 2 (IPv4 or IPv6) and with a SAFI of 73.

An update message that carries the MP_REACH_NLRI or MP_UNREACH_NLRI attribute with the SR Policy SAFI MUST also carry the BGP mandatory attributes. In addition, the BGP update message MAY also contain any of the BGP optional attributes.

The next-hop network address field in SR Policy SAFI (73) updates may be either a 4 octet IPv4 address or a 16 octet IPv6 address, independent of the SR Policy AFI. The length field of the next-hop address specifies the next-hop address family. If the next-hop length is 4, then the next-hop is an IPv4 address; if the next-hop length is 16, then it is a global IPv6 address; and if the next-hop length is 32, then it has a global IPv6 address followed by a link-local IPv6 address. The setting of the next-hop field and its attendant processing is governed by standard BGP procedures as described in section 3 in [RFC4760].

It is important to note that any BGP speaker receiving a BGP message with an SR Policy NLRI, will process it only if the NLRI is among the best-paths as per the BGP best-path selection algorithm. In other words, this document leverages the existing BGP propagation and best-path selection rules. Details of the procedures are described in Section 4.

It has to be noted that if several candidate paths of the same SR Policy (endpoint, color) are signaled via BGP to a head-end, it is RECOMMENDED that each NLRI use a different distinguisher. If BGP has installed into the BGP table two advertisements whose respective NLRIs have the same color and endpoint, but different distinguishers, both advertisements are passed to the SRPM as different candidate paths along with their respective originator information (i.e. ASN and BGP Router-ID) as described in section 2.4 of [I-D.ietf-spring-segment-routing-policy]. The ASN would be the ASN of origin and the BGP Router-ID is determined in the following order:

- From the Route Origin Community [RFC4360] if present and carrying an IP Address

- As the BGP Originator ID [RFC4456] if present

- As the BGP Router-ID of the peer from which the update was received as a last resort.

2.2. SR Policy and Tunnel Encapsulation Attribute

The content of the SR Policy Candidate Path is encoded in the Tunnel Encapsulation Attribute defined in [I-D.ietf-idr-tunnel-encaps] using a new Tunnel-Type called SR Policy Type with codepoint 15.

The SR Policy Encoding structure is as follows:

```
SR Policy SAFI NLRI: <Distinguisher, Policy-Color, Endpoint>
Attributes:
   Tunnel Encaps Attribute (23)
      Tunnel Type: SR Policy
          Binding SID
          SRv6 Binding SID
          Preference
          Priority
          Policy Name
          Policy Candidate Path Name
          Explicit NULL Label Policy (ENLP)
          Segment List
              Weight
              Segment
              Segment
              ...
          ...
```

where:

- SR Policy SAFI NLRI is defined in Section 2.1.

- Tunnel Encapsulation Attribute is defined in [I-D.ietf-idr-tunnel-encaps].

- Tunnel-Type is set to 15.

- Preference, Binding SID, SRv6 Binding SID, Priority, Policy Name, Policy Candidate Path Name, ENLP, Segment-List, Weight and Segment sub-TLVs are defined in this document.

- Additional sub-TLVs may be defined in the future.

A Tunnel Encapsulation Attribute MUST NOT contain more than one TLV of type "SR Policy".

2.3. Remote Endpoint and Color

The Remote Endpoint and Color sub-TLVs, as defined in [I-D.ietf-idr-tunnel-encaps], MAY also be present in the SR Policy encodings.

The Remote Endpoint and Color Sub-TLVs of the Tunnel Encapsulation Attribute are not used for SR Policy encodings and therefore their value is irrelevant in the context of the SR Policy SAFI NLRI. If present, the Remote Endpoint sub-TLV and the Color sub-TLV MUST be ignored by the BGP speaker.

2.4. SR Policy Sub-TLVs

This section specifies the sub-TLVs defined for encoding the information about the SR Policy Candidate Path.

Preference, Binding SID, SRv6 Binding SID, Segment-List, Priority, Policy Name, Policy Candidate Path Name and Explicit NULL Label Policy are the new sub-TLVs of the BGP Tunnel Encapsulation Attribute [I-D.ietf-idr-tunnel-encaps] being defined in this section.

Weight and Segment are sub-TLVs of the new Segment-List sub-TLV mentioned above.

None of the sub-TLVs defined in the following sub-sections have any effect on the BGP best-path selection or propagation procedures. These sub-TLVs are not used by BGP and are instead passed on to SRPM as SR Policy Candidate Path information for further processing described in [I-D.ietf-spring-segment-routing-policy] .

2.4.1. Preference Sub-TLV

The Preference sub-TLV is used to carry the preference of the SR Policy candidate path. The contents of this sub-TLV are used by the SRPM as described in section 2.7 in [I-D.ietf-spring-segment-routing-policy].

The Preference sub-TLV is optional and it MUST NOT appear more than once in the SR Policy encoding.

The Preference sub-TLV has following format:

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

where:

- Type: 12

- Length: 6.

- Flags: 1 octet of flags. None are defined at this stage. Flags SHOULD be set to zero on transmission and MUST be ignored on receipt.

- RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

- Preference: a 4-octet value.

2.4.2. Binding SID Sub-TLV

The Binding SID sub-TLV is used to signal the binding SID related information of the SR Policy candidate path. The contents of this sub-TLV are used by the SRPM as described in section 6 in [I-D.ietf-spring-segment-routing-policy].

The Binding SID sub-TLV is optional and it MUST NOT appear more than once in the SR Policy encoding.

When the Binding SID sub-TLV is used to signal an SRv6 SID, the choice of its SRv6 Endpoint Behavior [I-D.ietf-spring-srv6-network-programming] to be instantiated is left to the headend node. It is RECOMMENDED that the SRv6 Binding SID sub-TLV defined in Section 2.4.3, that enables the specification of the SRv6 Endpoint Behavior, be used for signaling of an SRv6 Binding SID for an SR Policy candidate path.

The Binding SID 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    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Binding SID (variable, optional)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 13

- Length: specifies the length of the value field not including Type and Length fields. Can be 2 or 6 or 18.

- Flags: 1 octet of flags. Following flags are defined in the new registry "SR Policy Binding SID Flags" as described in Section 6.6:

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

where:

* S-Flag: This flag encodes the "Specified-BSID-only" behavior. It is used by SRPM as described in section 6.2.3 in [I-D.ietf-spring-segment-routing-policy].

* I-Flag: This flag encodes the "Drop Upon Invalid" behavior. It is used by SRPM as described in section 8.2 in [I-D.ietf-spring-segment-routing-policy].

* Unused bits in the Flag octet SHOULD be set to zero upon transmission and MUST be ignored upon receipt.

o RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

o Binding SID: if length is 2, then no Binding SID is present. If length is 6 then the Binding SID is encoded in 4 octets using the format below. TC, S, TTL (Total of 12 bits) are RESERVED and SHOULD be set to zero and MUST be ignored.

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

If length is 18 then the Binding SID contains a 16-octet SRv6 SID.

2.4.3. SRv6 Binding SID Sub-TLV

The SRv6 Binding SID sub-TLV is used to signal the SRv6 Binding SID related information of the SR Policy candidate path. It enables the specification of the SRv6 Endpoint Behavior [I-D.ietf-spring-srv6-network-programming] to be instantiated on the headend node. The contents of this sub-TLV are used by the SRPM as described in section 6 in [I-D.ietf-spring-segment-routing-policy].

The SRv6 Binding SID sub-TLV is optional. More than one SRv6 Binding SIDs MAY be signalled in the same SR Policy encoding to indicate one or more SRv6 SIDs, each with potentially different SRv6 Endpoint Behaviors to be instantiated.

The SRv6 Binding SID 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    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                 SRv6 Binding SID (16 octets)                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //     SRv6 Endpoint Behavior and SID Structure (optional)     //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

o Type: TBD

o Length is variable

o Flags: 1 octet of flags. Following flags are defined in the new registry "SR Policy Binding SID Flags" as described in Section 6.7:

```
    0 1 2 3 4 5 6 7
   +-+-+-+-+-+-+-+-+
   |S|I|B|         |
   +-+-+-+-+-+-+-+-+
```

where:

* S-Flag: This flag encodes the "Specified-BSID-only" behavior. It is used by SRPM as described in section 6.2.3 in [I-D.ietf-spring-segment-routing-policy].

* I-Flag: This flag encodes the "Drop Upon Invalid" behavior. It is used by SRPM as described in section 8.2 in [I-D.ietf-spring-segment-routing-policy].

* B-Flag: This flag, when set, indicates the presence of the SRv6 Endpoint Behavior and SID Structure encoding specified in Section 2.4.4.2.13.

* Unused bits in the Flag octet SHOULD be set to zero upon transmission and MUST be ignored upon receipt.

o RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

o SRv6 Binding SID: Contains a 16-octet SRv6 SID.

- SRv6 Endpoint Behavior and SID Structure: Optional, as defined in Section 2.4.4.2.13.

2.4.4. Segment List Sub-TLV

The Segment List sub-TLV encodes a single explicit path towards the endpoint as described in section 5.1 in [I-D.ietf-spring-segment-routing-policy]. The Segment List sub-TLV includes the elements of the paths (i.e., segments) as well as an optional Weight sub-TLV.

The Segment List sub-TLV may exceed 255 bytes length due to large number of segments. Therefore a 2-octet length is required. According to [I-D.ietf-idr-tunnel-encaps], the first bit of the sub-TLV codepoint defines the size of the length field. Therefore, for the Segment List sub-TLV a code point of 128 or higher is used.

The Segment List sub-TLV is optional and MAY appear multiple times in the SR Policy encoding. The ordering of Segment List sub-TLVs, each sub-TLV encoding a Segment List, does not matter.

The Segment List sub-TLV contains zero or more Segment sub-TLVs and MAY contain a Weight sub-TLV.

The Segment List 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            |   RESERVED    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //                            sub-TLVs                          //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 128.

- Length: the total length (not including the Type and Length fields) of the sub-TLVs encoded within the Segment List sub-TLV.

- RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

- sub-TLVs currently defined:

  * An optional single Weight sub-TLV.

- Zero or more Segment sub-TLVs.

Validation of an explicit path encoded by the Segment List sub-TLV is beyond the scope of BGP and performed by the SRPM as described in section 5 in [I-D.ietf-spring-segment-routing-policy].

2.4.4.1. Weight Sub-TLV

The Weight sub-TLV specifies the weight associated to a given segment list. The contents of this sub-TLV are used only by the SRPM as described in section 2.11 in [I-D.ietf-spring-segment-routing-policy].

The Weight sub-TLV is optional and it MUST NOT appear more than once inside the Segment List sub-TLV.

The Weight 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    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Weight                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 9.

- Length: 6

- Flags: 1 octet of flags. None are defined at this stage. Flags SHOULD be set to zero on transmission and MUST be ignored on receipt.

- RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

2.4.4.2. Segment Sub-TLVs

A Segment sub-TLV describes a single segment in a segment list (i.e., a single element of the explicit path). One or more Segment sub-TLVs constitute an explicit path of the SR Policy candidate path. The contents of these sub-TLVs are used only by the SRPM as described in section 4 in [I-D.ietf-spring-segment-routing-policy].

The Segment sub-TLVs are optional and MAY appear multiple times in the Segment List sub-TLV.

[I-D.ietf-spring-segment-routing-policy] defines several Segment Types:

```
 Type  A: SID only, in the form of MPLS Label
 Type  B: SID only, in the form of IPv6 address
 Type  C: IPv4 Node Address with optional SID
 Type  D: IPv6 Node Address with optional SID for SR MPLS
 Type  E: IPv4 Address and index with optional SID
 Type  F: IPv4 Local and Remote addresses with optional SID
 Type  G: IPv6 Address and index for local and remote pair with optional
          SID for SR MPLS
 Type  H: IPv6 Local and Remote addresses with optional SID for SR MPLS
 Type  I: IPv6 Node Address with optional SID for SRv6
 Type  J: IPv6 Address and index for local and remote pair with optional
          SID for SRv6
 Type  K: IPv6 Local and Remote addresses for SRv6
```

The follow sub-sections specify the sub-TLV used for encoding each of these Segment Types.

2.4.4.2.1. Type A: SID only, in the form of MPLS Label

The Type A Segment Sub-TLV encodes a single SR-MPLS SID. The format is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |     Flags     |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Label                        | TC  |S|      TTL      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- o Type: 1.

- o Length is 6.

- o Flags: 1 octet of flags as defined in Section 2.4.4.2.12.

- o RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

- o Label: 20 bits of label value.

- o TC: 3 bits of traffic class.

- o S: 1 bit of bottom-of-stack.

- o TTL: 1 octet of TTL.

The following applies to the Type-1 Segment sub-TLV:

- o The S bit SHOULD be zero upon transmission, and MUST be ignored upon reception.

- o If the originator wants the receiver to choose the TC value, it sets the TC field to zero.

- o If the originator wants the receiver to choose the TTL value, it sets the TTL field to 255.

- o If the originator wants to recommend a value for these fields, it puts those values in the TC and/or TTL fields.

- o The receiver MAY override the originator's values for these fields. This would be determined by local policy at the receiver. One possible policy would be to override the fields only if the fields have the default values specified above.

2.4.4.2.2. Type B: SID only, in the form of IPv6 address

The Type B Segment Sub-TLV encodes a single SRv6 SID. The format is as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |    Length     |     Flags     |   RESERVED    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //                      SRv6 SID (16 octets)                    //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //     SRv6 Endpoint Behavior and SID Structure (optional)      //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- o Type: 13.

- o Length is variable.

- Flags: 1 octet of flags as defined in Section 2.4.4.2.12.

- RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

- SRv6 SID: 16 octets of IPv6 address.

- SRv6 Endpoint Behavior and SID Structure: Optional, as defined in Section 2.4.4.2.13.

The TLV 2 defined for advertisement of Segment Type B in the earlier versions of this document has been deprecated to avoid backward compatibility issues (refer Appendix A for details).

2.4.4.2.3. Type C: IPv4 Node Address with optional SID

The Type C Segment Sub-TLV encodes an IPv4 node address, SR Algorithm and an optional SR-MPLS SID. The format is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Length      |     Flags     | SR Algorithm  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 IPv4 Node Address (4 octets)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                SR-MPLS SID (optional, 4 octets)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 3.

- Length is 10 when the SR-MPLS SID is present else is 6.

- Flags: 1 octet of flags as defined in Section 2.4.4.2.12.

- SR Algorithm: 1 octet specifying SR Algorithm as described in section 3.1.1 in [RFC8402], when A-Flag as defined in Section 2.4.4.2.12 is present. SR Algorithm is used by SRPM as described in section 4 in [I-D.ietf-spring-segment-routing-policy]. When A-Flag is not encoded, this field SHOULD be set to zero on transmission and MUST be ignored on receipt.

- IPv4 Node Address: a 4 octet IPv4 address representing a node.

- o SR-MPLS SID: optional, 4 octet field containing label, TC, S and TTL as defined in Section 2.4.4.2.1.

2.4.4.2.4. Type D: IPv6 Node Address with optional SID for SR MPLS

The Type D Segment Sub-TLV encodes an IPv6 node address, SR Algorithm and an optional SR-MPLS SID. The format is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Length      |     Flags     | SR Algorithm  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//               IPv6 Node Address (16 octets)                  //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                SR-MPLS SID (optional, 4 octets)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- o Type: 4

- o Length is 22 when the SR-MPLS SID is present else is 18.

- o Flags: 1 octet of flags as defined in Section 2.4.4.2.12.

- o SR Algorithm: 1 octet specifying SR Algorithm as described in section 3.1.1 in [RFC8402], when A-Flag as defined in Section 2.4.4.2.12 is present. SR Algorithm is used by SRPM as described in section 4 in [I-D.ietf-spring-segment-routing-policy]. When A-Flag is not encoded, this field SHOULD be set to zero on transmission and MUST be ignored on receipt.

- o IPv6 Node Address: a 16 octet IPv6 address representing a node.

- o SR-MPLS SID: optional, 4 octet field containing label, TC, S and TTL as defined in Section 2.4.4.2.1.

2.4.4.2.5. Type E: IPv4 Address + Local Interface ID with optional SID

The Type E Segment Sub-TLV encodes an IPv4 node address, a local interface Identifier (Local Interface ID) and an optional SR-MPLS SID. The format is as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |   Length      |     Flags     |   RESERVED    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                 Local Interface ID (4 octets)                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                 IPv4 Node Address (4 octets)                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                SR-MPLS SID (optional, 4 octets)               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 5.

- Length is 14 when the SR-MPLS SID is present else is 10.

- Flags: 1 octet of flags as defined in Section 2.4.4.2.12.

- RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

- Local Interface ID: 4 octets of interface index as defined in [RFC8664].

- IPv4 Node Address: a 4 octet IPv4 address representing a node.

- SR-MPLS SID: optional, 4 octet field containing label, TC, S and TTL as defined in Section 2.4.4.2.1.

2.4.4.2.6. Type F: IPv4 Local and Remote addresses with optional SID

The Type F Segment Sub-TLV encodes an adjacency local address, an adjacency remote address and an optional SR-MPLS SID. The format is as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |   Length      |     Flags     |   RESERVED    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                Local IPv4 Address (4 octets)                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                Remote IPv4 Address  (4 octets)                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                SR-MPLS SID (optional, 4 octets)               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 6.

- Length is 14 when the SR-MPLS SID is present else is 10.

- Flags: 1 octet of flags as defined in Section 2.4.4.2.12.

- RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

- Local IPv4 Address: a 4 octet IPv4 address.

- Remote IPv4 Address: a 4 octet IPv4 address.

- SR-MPLS SID: optional, 4 octet field containing label, TC, S and TTL as defined in Section 2.4.4.2.1.

2.4.4.2.7. Type G: IPv6 Address + Interface ID for local and remote pair with optional SID for SR MPLS

The Type G Segment Sub-TLV encodes an IPv6 Link Local adjacency with IPv6 local node address, a local interface identifier (Local Interface ID), IPv6 remote node address , a remote interface identifier (Remote Interface ID) and an optional SR-MPLS SID. The format is as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |   Length      |     Flags     |   RESERVED    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                 Local Interface ID (4 octets)                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //                IPv6 Local Node Address (16 octets)          //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                 Remote Interface ID (4 octets)                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //                IPv6 Remote Node Address (16 octets)         //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                SR-MPLS SID (optional, 4 octets)               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 7

- Length is 46 when the SR-MPLS SID is present else is 42.

- Flags: 1 octet of flags as defined in Section 2.4.4.2.12.

- RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

- Local Interface ID: 4 octets of interface index as defined in [RFC8664].

- IPv6 Local Node Address: a 16 octet IPv6 address.

- Remote Interface ID: 4 octets of interface index as defined in [RFC8664]. The value MAY be set to zero when the local node address and interface identifiers are sufficient to describe the link.

- IPv6 Remote Node Address: a 16 octet IPv6 address. The value MAY be set to zero when the local node address and interface identifiers are sufficient to describe the link.

- SR-MPLS SID: optional, 4 octet field containing label, TC, S and TTL as defined in Section 2.4.4.2.1.

2.4.4.2.8. Type H: IPv6 Local and Remote addresses with optional SID for SR MPLS

The Type H Segment Sub-TLV encodes an adjacency local address, an adjacency remote address and an optional SR-MPLS SID. The format is as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |   Length      |     Flags     |   RESERVED    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //               Local IPv6 Address (16 octets)                //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //               Remote IPv6 Address  (16 octets)              //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                SR-MPLS SID (optional, 4 octets)               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 8
- Length is 38 when the SR-MPLS SID is present else is 34.
- Flags: 1 octet of flags as defined in Section 2.4.4.2.12.
- RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.
- Local IPv6 Address: a 16 octet IPv6 address.
- Remote IPv6 Address: a 16 octet IPv6 address.
- SR-MPLS SID: optional, 4 octet field containing label, TC, S and TTL as defined in Section 2.4.4.2.1.

2.4.4.2.9. Type I: IPv6 Node Address with optional SRv6 SID

The Type I Segment Sub-TLV encodes an IPv6 node address, SR Algorithm and an optional SRv6 SID. The format is as follows:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |   Length      |     Flags     | SR Algorithm  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //                 IPv6 Node Address (16 octets)               //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //                    SRv6 SID (optional, 16 octets)           //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //     SRv6 Endpoint Behavior and SID Structure (optional)     //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 14

- Length is variable.

- Flags: 1 octet of flags as defined in Section 2.4.4.2.12.

- SR Algorithm: 1 octet specifying SR Algorithm as described in section 3.1.1 in [RFC8402], when A-Flag as defined in Section 2.4.4.2.12 is present. SR Algorithm is used by SRPM as described in section 4 in [I-D.ietf-spring-segment-routing-policy]. When A-Flag is not encoded, this field SHOULD be set to zero on transmission and MUST be ignored on receipt.

- IPv6 Node Address: a 16 octet IPv6 address.

- SRv6 SID: optional, 16 octet IPv6 address.

- SRv6 Endpoint Behavior and SID Structure: Optional, as defined in Section 2.4.4.2.13.

The TLV 10 defined for advertisement of Segment Type I in the earlier versions of this document has been deprecated to avoid backward compatibility issues (refer Appendix A for details).

2.4.4.2.10. Type J: IPv6 Address + Interface ID for local and remote pair for SRv6 with optional SID

The Type J Segment Sub-TLV encodes an IPv6 Link Local adjacency with local node address, a local interface identifier (Local Interface ID), remote IPv6 node address, a remote interface identifier (Remote Interface ID) and an optional SRv6 SID. The format is as follows:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |   Length      |     Flags     | SR Algorithm  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                 Local Interface ID (4 octets)                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //                IPv6 Local Node Address (16 octets)          //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                 Remote Interface ID (4 octets)                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //                IPv6 Remote Node Address (16 octets)         //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //                SRv6 SID (optional, 16 octets)               //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //     SRv6 Endpoint Behavior and SID Structure (optional)     //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 15

- Length is variable.

- Flags: 1 octet of flags as defined in Section 2.4.4.2.12.

- SR Algorithm: 1 octet specifying SR Algorithm as described in section 3.1.1 in [RFC8402], when A-Flag as defined in Section 2.4.4.2.12 is present. SR Algorithm is used by SRPM as described in section 4 in [I-D.ietf-spring-segment-routing-policy]. When A-Flag is not encoded, this field SHOULD be set to zero on transmission and MUST be ignored on receipt.

- Local Interface ID: 4 octets of interface index as defined in [RFC8664].

- IPv6 Local Node Address: a 16 octet IPv6 address.

- Remote Interface ID: 4 octets of interface index as defined in [RFC8664]. The value MAY be set to zero when the local node address and interface identifiers are sufficient to describe the link.

- IPv6 Remote Node Address: a 16 octet IPv6 address. The value MAY be set to zero when the local node address and interface identifiers are sufficient to describe the link.

- SRv6 SID: optional, 16 octet IPv6 address.

- SRv6 Endpoint Behavior and SID Structure: Optional, as defined in Section 2.4.4.2.13.

The TLV 11 defined for advertisement of Segment Type J in the earlier versions of this document has been deprecated to avoid backward compatibility issues (refer Appendix A for details).

2.4.4.2.11. Type K: IPv6 Local and Remote addresses for SRv6 with optional SID

The Type K Segment Sub-TLV encodes an adjacency local address, an adjacency remote address and an optional SRv6 SID. The format is as follows:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |     Length    |     Flags     | SR Algorithm  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //               Local IPv6 Address (16 octets)                //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //               Remote IPv6 Address  (16 octets)              //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //                SRv6 SID (optional, 16 octets)               //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //     SRv6 Endpoint Behavior and SID Structure (optional)     //
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 16

- Length is variable.

- Flags: 1 octet of flags as defined in Section 2.4.4.2.12.

- SR Algorithm: 1 octet specifying SR Algorithm as described in section 3.1.1 in [RFC8402], when A-Flag as defined in Section 2.4.4.2.12 is present. SR Algorithm is used by SRPM as described in section 4 in [I-D.ietf-spring-segment-routing-policy]. When A-Flag is not encoded, this field SHOULD be set to zero on transmission and MUST be ignored on receipt.

- Local IPv6 Address: a 16 octet IPv6 address.

- Remote IPv6 Address: a 16 octet IPv6 address.

- SRv6 SID: optional, 16 octet IPv6 address.

- SRv6 Endpoint Behavior and SID Structure: Optional, as defined in Section 2.4.4.2.13.

The TLV 12 defined for advertisement of Segment Type K in the earlier versions of this document has been deprecated to avoid backward compatibility issues (refer Appendix A for details).

2.4.4.2.12. Segment Flags

The Segment Types sub-TLVs described above MAY contain following flags in the "Flags" field defined in Section 6.8:

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

where:

V-Flag: This flag, when set, is used by SRPM for the purpose of "SID verification" as described in Section 5.1 in [I-D.ietf-spring-segment-routing-policy].

A-Flag: This flag, when set, indicates the presence of SR Algorithm id in the "SR Algorithm" field applicable to various Segment Types. SR Algorithm is used by SRPM as described in section 4 in [I-D.ietf-spring-segment-routing-policy].

S-Flag: This flag, when set, indicates the presence of the SR-MPLS or SRv6 SID depending on the segment type.

B-Flag: This flag, when set, indicates the presence of the SRv6 Endpoint Behavior and SID Structure encoding specified in Section 2.4.4.2.13.

Unused bits in the Flag octet SHOULD be set to zero upon transmission and MUST be ignored upon receipt.

The following applies to the Segment Flags:

- V-Flag is applicable to all Segment Types.

- A-Flag is applicable to Segment Types C, D, I, J and K. If A-Flag appears with Segment Types A, B, E, F, G and H, it MUST be ignored.

- S-Flag is applicable to Segment Types C, D, E, F, G, H, I, J and K. If S-Flag appears with Segment Types A or B, it MUST be ignored.

- B-Flag is applicable to Segment Types B, I, J and K. If B-Flag appears with Segment Types A, C, D, E, F, G and H, it MUST be ignored.

2.4.4.2.13. SRv6 SID Endpoint Behavior and Structure

The Segment Types sub-TLVs described above MAY contain the SRv6 Endpoint Behavior and SID Structure [I-D.ietf-spring-srv6-network-programming] encoding as described below:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Endpoint Behavior       |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    LB Length  |  LN Length    | Fun. Length   |  Arg. Length  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Endpoint Behavior: 2 octets. The SRv6 Endpoint Behavior code point for this SRv6 SID as defined in section 9.2 of [I-D.ietf-spring-srv6-network-programming]. When set with the value 0, the choice of SRv6 Endpoint Behavior is left to the headend.

Reserved: 2 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

Locator Block Length: 1 octet. SRv6 SID Locator Block length in bits.

Locator Node Length: 1 octet. SRv6 SID Locator Node length in bits.

Function Length: 1 octet. SRv6 SID Function length in bits.

Argument Length: 1 octet. SRv6 SID Arguments length in bits.

The total of the locator block, locator node, function and argument lengths MUST be less than or equal to 128.

2.4.5. Explicit NULL Label Policy Sub-TLV

In order to steer an unlabeled IP packet into an SR policy, it is necessary to create a label stack for that packet, and to push one or more labels onto that stack.

The Explicit NULL Label Policy (ENLP) sub-TLV is used to indicate whether an Explicit NULL Label [RFC3032] must be pushed on an unlabeled IP packet before any other labels.

If an ENLP Sub-TLV is not present, the decision of whether to push an Explicit NULL label on a given packet is a matter of local configuration.

The ENLP sub-TLV is optional and it MUST NOT appear more than once in the SR Policy encoding.

The contents of this sub-TLV are used by the SRPM as described in section 4.1 in [I-D.ietf-spring-segment-routing-policy].

```
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    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     ENLP      |
+-+-+-+-+-+-+-+-+
```

Where:

Type: 14.

Length: 3.

Flags: 1 octet of flags. None are defined at this stage. Flags SHOULD be set to zero on transmission and MUST be ignored on receipt.

RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

ENLP (Explicit NULL Label Policy): Indicates whether Explicit NULL labels are to be pushed on unlabeled IP packets that are being steered into a given SR policy. This field has one of the following values:

0: Reserved.

1: Push an IPv4 Explicit NULL label on an unlabeled IPv4 packet, but do not push an IPv6 Explicit NULL label on an unlabeled IPv6 packet.

2: Push an IPv6 Explicit NULL label on an unlabeled IPv6 packet, but do not push an IPv4 Explicit NULL label on an unlabeled IPv4 packet.

3: Push an IPv4 Explicit NULL label on an unlabeled IPv4 packet, and push an IPv6 Explicit NULL label on an unlabeled IPv6 packet.

4: Do not push an Explicit NULL label.

5 - 255: Reserved.

The ENLP reserved values may be used for future extensions and implementations SHOULD ignore the ENLP Sub-TLV with these values. The behavior signaled in this Sub-TLV MAY be overridden by local configuration. The section 4.1 of [I-D.ietf-spring-segment-routing-policy] draft describes the behavior on the headend for handling of explicit null label.

2.4.6. Policy Priority Sub-TLV

An operator MAY set the Policy Priority sub-TLV to indicate the order in which the SR policies are re-computed upon topological change. The contents of this sub-TLV are used by the SRPM as described in section 2.11 in [I-D.ietf-spring-segment-routing-policy].

The Priority sub-TLV is optional and it MUST NOT appear more than once in the SR Policy encoding.

The Priority sub-TLV has following format:

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

Where:

Type: 15

Length: 2.

Priority: a 1-octet value.

RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

2.4.7. Policy Candidate Path Name Sub-TLV

An operator MAY set the Policy Candidate Path Name sub-TLV to attach a symbolic name to the SR Policy candidate path.

Usage of Policy Candidate Path Name sub-TLV is described in section 2.6 in [I-D.ietf-spring-segment-routing-policy].

The Policy Candidate Path Name sub-TLV may exceed 255 bytes length due to long name. Therefore a 2-octet length is required. According to [I-D.ietf-idr-tunnel-encaps], the first bit of the sub-TLV codepoint defines the size of the length field. Therefore, for the Policy Candidate Path Name sub-TLV a code point of 128 or higher is used.

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.

The Policy Candidate Path Name sub-TLV is optional and it MUST NOT appear more than once in the SR Policy encoding.

The Policy Candidate Path Name sub-TLV has following format:

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

Where:

Type: 129.

Length: Variable.

RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

Policy Candidate Path Name: Symbolic name for the SR Policy candidate path without a NULL terminator as specified in section 2.6 of [I-D.ietf-spring-segment-routing-policy].

2.4.8. Policy Name Sub-TLV

An operator MAY set the Policy Name sub-TLV to associate a symbolic name with the SR Policy for which the candidate path is being advertised via the SR Policy NLRI.

Usage of Policy Name sub-TLV is described in section 2.1 of [I-D.ietf-spring-segment-routing-policy].

The Policy Name sub-TLV may exceed 255 bytes length due to long policy name. Therefore a 2-octet length is required. According to [I-D.ietf-idr-tunnel-encaps], the first bit of the sub-TLV codepoint defines the size of the length field. Therefore, for the Policy Name sub-TLV a code point of 128 or higher is used.

The Policy Name sub-TLV is optional and it MUST NOT appear more than once in the SR Policy encoding.

The Policy Name sub-TLV has following format:

```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Length                      |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                        Policy Name                          //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

Type: TBD

Length: Variable.

RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

Policy Name: Symbolic name for the policy. It SHOULD be a string of printable ASCII characters, without a NULL terminator.

3. Color Extended Community

The Color Extended Community as defined in [I-D.ietf-idr-tunnel-encaps] is used to steer traffic into a policy.

When the Color Extended Community is used for the purpose of steering the traffic into an SR Policy, two bits from the Flags field (as defined in [I-D.ietf-idr-tunnel-encaps]) are used as follows:

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

where CO bits are defined as the "Color-Only" bits. [I-D.ietf-spring-segment-routing-policy] defines the influence of these bits on the automated steering of BGP Payload traffic onto SR Policies.

4. SR Policy Operations

As described in this document, BGP is not the actual consumer of an SR Policy NLRI. BGP is in charge of the origination and propagation of the SR Policy NLRI but its installation and use is outside the scope of BGP. The details of SR Policy installation and use are specified in [I-D.ietf-spring-segment-routing-policy].

4.1. Advertisement of SR Policies

Typically, but not limited to, an SR Policy is computed by a controller or a path computation engine (PCE) and originated by a BGP speaker on its behalf.

Multiple SR Policy NLRIs may be present with the same <color, endpoint> tuple but with different content when these SR policies are intended for different head-ends.

The distinguisher of each SR Policy NLRI prevents undesired BGP route selection among these SR Policy NLRIs and allows their propagation across route reflectors [RFC4456].

Moreover, one or more route-target SHOULD be attached to the advertisement, where each route-target identifies one or more intended head-ends for the advertised SR Policy update.

If no route-target is attached to the SR Policy NLRI, then it is assumed that the originator sends the SR Policy update directly (e.g., through a BGP session) to the intended receiver. In such case, the NO_ADVERTISE community MUST be attached to the SR Policy update.

4.2. Reception of an SR Policy NLRI

On reception of an SR Policy NLRI, a BGP speaker first determines if it is acceptable and then if it is usable.

4.2.1. Acceptance of an SR Policy NLRI

When a BGP speaker receives an SR Policy NLRI from a neighbor it MUST first determine if it's acceptable. The following rules apply in addition to the validation described in Section 5:

- o The SR Policy NLRI MUST include a distinguisher, color and endpoint field which implies that the length of the NLRI MUST be either 12 or 24 octets (depending on the address family of the endpoint).

- o The SR Policy update MUST have either the NO_ADVERTISE community or at least one route-target extended community in IPv4-address format or both. If a router supporting this specification receives an SR Policy update with no route-target extended communities and no NO_ADVERTISE community, the update MUST be considered as malformed.

- o The Tunnel Encapsulation Attribute MUST be attached to the BGP Update and MUST have a Tunnel Type TLV set to SR Policy (codepoint is 15).

A router that receives an SR Policy update that is not valid according to these criteria MUST treat the update as malformed and the SR Policy candidate path MUST NOT be passed to the SRPM.

4.2.2. Usable SR Policy NLRI

A SR Policy update that has been determined to be acceptable is further evaluated for its usability by the receiving node.

An SR Policy NLRI update without any route-target extended community but having the NO_ADVERTISE community is considered usable.

If one or more route-targets are present, then at least one route-target MUST match the BGP Identifier of the receiver for the update to be considered usable. The BGP Identifier is defined in [RFC4271] as a 4 octet IPv4 address. Therefore, the route-target extended community MUST be of the same format.

If one or more route-targets are present and none matches the local BGP Identifier, then, while the SR Policy NLRI is acceptable, it is not usable on the receiver node.

When the SR Policy tunnel type includes any sub-TLV that is unrecognized or unsupported, the update SHOULD NOT be considered usable. An implementation MAY provide an option for ignoring unsupported sub-TLVs.

4.2.3. Passing a usable SR Policy NLRI to the SRPM

Once BGP on the receiving node has determined that the SR Policy NLRI is usable, it passes the SR Policy candidate path to the SRPM. Note that, along with the candidate path details, BGP also passes the originator information for breaking ties in the candidate path selection process as described in section 2.4 in [I-D.ietf-spring-segment-routing-policy].

When an update for an SR Policy NLRI results in it’s becoming unusable, BGP MUST delete it’s corresponding SR Policy candidate path from the SRPM.

The SRPM applies the rules defined in section 2 in [I-D.ietf-spring-segment-routing-policy] to determine whether the SR Policy candidate path is valid and to select the best candidate path among the valid ones for a given SR Policy.

4.2.4. Propagation of an SR Policy

SR Policy NLRIs that have been determined acceptable and valid can be evaluated for propagation, even the ones that are not usable.

SR Policy NLRIs that have the NO_ADVERTISE community attached to them MUST NOT be propagated.

By default, a BGP node receiving an SR Policy NLRI MUST NOT propagate it to any EBGP neighbor. An implementation MAY provide an explicit configuration to override this and enable propagation of acceptable SR Policy NLRIs to specific EBGP neighbors.

A BGP node advertises a received SR Policy NLRI to its IBGP neighbors according to normal IBGP propagation rules.

By default, a BGP node receiving an SR Policy NLRI SHOULD NOT remove route-target extended community before propagation. An implementation MAY provide support for configuration to filter and/or remove route-target extended community before propagation.

5. Error Handling

This section describes the error handling actions, as described in [RFC7606], that are to be performed for handling of BGP update messages for BGP SR Policy SAFI.

A BGP Speaker MUST perform the following syntactic validation of the SR Policy NLRI to determine if it is malformed. This includes the validation of length of each NLRI and the total length of the MP_REACH_NLRI and MP_UNREACH_NLRI attributes.

When the error determined allows for the router to skip the malformed NLRI(s) and continue processing of the rest of the update message, then it MUST handle such malformed NLRIs as 'Treat-as-withdraw'. In other cases, where the error in the NLRI encoding results in the inability to process the BGP update message (e.g. length related encoding errors), then the router SHOULD handle such malformed NLRIs as 'AFI/SAFI disable' when other AFI/SAFI besides SR Policy are being advertised over the same session. Alternately, the router MUST perform 'session reset' when the session is only being used for SR Policy or when it 'AFI/SAFI disable' action is not possible.

The validation of the TLVs/sub-TLVs introduced in this document and defined in their respective sub-sections of Section 2.4 MUST be performed to determine if they are malformed or invalid. The validation of the Tunnel Encapsulation Attribute itself and the other TLVs/sub-TLVs specified in [I-D.ietf-idr-tunnel-encaps] MUST be done as described in that document. In case of any error detected, either at the attribute or its TLV/sub-TLV level, the "treat-as-withdraw" strategy MUST be applied. This is because an SR Policy update without a valid Tunnel Encapsulation Attribute (comprising of all valid TLVs/sub-TLVs) is not usable.

An SR Policy update that is determined to be not acceptable, and therefore malformed, based on rules described in Section 4.2.1 MUST be handled by the "treat-as-withdraw" strategy.

The validation of the individual fields of the TLVs/sub-TLVs defined in Section 2.4 are beyond the scope of BGP as they are handled by the SRPM as described in the individual TLV/sub-TLV sub-sections. A BGP implementation MUST NOT perform semantic verification of such fields nor consider the SR Policy update to be invalid or not acceptable/usable on the basis of such a validation.

An implementation SHOULD log an error for any errors found during the above validation for further analysis.

6. IANA Considerations

This document requests codepoint allocations for new TLVs/sub-TLVs in following existing registries:

- Subsequent Address Family Identifiers (SAFI) Parameters registry

- BGP Tunnel Encapsulation Attribute Tunnel Types registry under the BGP Parameters registry

- BGP Tunnel Encapsulation Attribute sub-TLVs registry under the BGP Parameters registry

- Color Extended Community Flags registry under the BGP Extended Communities registry

This document also requests creation of the following new registries:

- SR Policy Segment List Sub-TLVs under the BGP Parameters registry

- SR Policy Binding SID Flags under the BGP Parameters registry

- SR Policy Segment Flags under the BGP Parameters registry

6.1. Existing Registry: Subsequent Address Family Identifiers (SAFI) Parameters

This document defines a new SAFI in the registry "Subsequent Address Family Identifiers (SAFI) Parameters" that has been assigned a codepoint by IANA as follows:

Codepoint	Description	Reference
73	SR Policy SAFI	This document

6.2. Existing Registry: BGP Tunnel Encapsulation Attribute Tunnel Types

This document defines a new Tunnel-Type in the registry "BGP Tunnel Encapsulation Attribute Tunnel Types" that has been assigned a codepoint by IANA as follows:

Codepoint	Description	Reference
15	SR Policy Type	This document

6.3. Existing Registry: BGP Tunnel Encapsulation Attribute sub-TLVs

This document defines new sub-TLVs in the registry "BGP Tunnel Encapsulation Attribute sub-TLVs" that has been assigned codepoints by IANA as follows:

Codepoint	Description	Reference
12	Preference sub-TLV	This document
13	Binding SID sub-TLV	This document
14	ENLP sub-TLV	This document
15	Priority sub-TLV	This document
TBD	SRv6 Binding SID sub-TLV	This document
128	Segment List sub-TLV	This document
129	Policy CP Name sub-TLV	This document
TBD	Policy Name sub-TLV	This document

6.4. Existing Registry: Color Extended Community Flags

This document requests allocations in the registry called "Color Extended Community Flags" under the "BGP Tunnel Encapsulation Parameters" grouping.

Following bits are to be allocated:

Bit Position	Description	Reference
0-1	Color-only bits	This document

6.5. New Registry: SR Policy Segment List Sub-TLVs

This document requests creation of a new registry called "SR Policy Segment List Sub-TLVs". The allocation policy of this registry is "Specification Required" according to [RFC8126].

Following initial Sub-TLV codepoints are assigned by this document:

Value	Description	Reference
1	Type A MPLS SID sub-TLV	This document
2	Deprecated	This document
3	Type C IPv4 Node and SID sub-TLV	This document
4	Type D IPv6 Node and SID for SR-MPLS sub-TLV	This document
5	Type E IPv4 Node, index and SID sub-TLV	This document
6	Type F IPv4 Local/Remote addresses and SID sub-TLV	This document
7	Type G IPv6 Node, index for remote and local pair and SID for SR-MPLS sub-TLV	This document
8	Type H IPv6 Local/Remote addresses and SID sub-TLV	This document
9	Weight sub-TLV	This document
10	Deprecated	This document
11	Deprecated	This document
12	Deprecated	This document
13	Type B SRv6 SID sub-TLV	This document
14	Type I IPv6 Node and SID for SRv6 sub-TLV	This document
15	Type J IPv6 Node, index for remote and local pair and SID for SRv6 sub-TLV	This document
16	Type K IPv6 Local/Remote addresses and SID for SRv6 sub-TLV	This document

6.6. New Registry: SR Policy Binding SID Flags

This document requests creation of a new registry called "SR Policy Binding SID Flags". The allocation policy of this registry is "Specification Required" according to [RFC8126].

Following flags are defined:

Bit	Description	Reference
0	Specified-BSID-Only Flag (S-Flag)	This document
1	Drop Upon Invalid Flag (I-Flag)	This document
2-7	Unassigned	

6.7. New Registry: SR Policy SRv6 Binding SID Flags

This document requests creation of a new registry called "SR Policy SRv6 Binding SID Flags". The allocation policy of this registry is "Specification Required" according to [RFC8126].

Following flags are defined:

Bit	Description	Reference
0	Specified-BSID-Only Flag (S-Flag)	This document
1	Drop Upon Invalid Flag (I-Flag)	This document
2	SRv6 Endpoint Behavior & SID Structure Flag (B-Flag)	This document
3-7	Unassigned	

6.8. New Registry: SR Policy Segment Flags

This document requests creation of a new registry called "SR Policy Segment Flags". The allocation policy of this registry is "Specification Required" according to [RFC8126].

Following Flags are defined:

Bit	Description	Reference
0	Segment Verification Flag (V-Flag)	This document
1	SR Algorithm Flag (A-Flag)	This document
2	SID Specified Flag (S-Flag)	This document
3	SRv6 Endpoint Behavior & SID Structure Flag (B-Flag)	This document
4-7	Unassigned	

6.9. Guidance for Designated Experts

In all cases of review by the Designated Expert (DE) described here, the DE is expected to ascertain the existence of suitable documentation (a specification) as described in [RFC8126]. The DE is also expected to check the clarity of purpose and use of the requested code points. Additionally, the DE must verify that any request for one of these code points has been made available for review and comment within the IETF: the DE will post the request to the IDR Working Group mailing list (or a successor mailing list designated by the IESG). If the request comes from within the IETF, it should be documented in an Internet-Draft. Lastly, the DE must ensure that any other request for a code point does not conflict with work that is active or already published within the IETF.

7. Security Considerations

The security mechanisms of the base BGP security model apply to the extensions described in this document as well. See the Security Considerations section of [RFC4271] for a discussion of BGP security. Also refer to [RFC4272] and [RFC6952] for analysis of security issues for BGP.

The BGP SR Policy extensions specified in this document enable traffic engineering and service programming use-cases within the SR domain as described in [I-D.ietf-spring-segment-routing-policy] . SR operates within a trusted SR domain [RFC8402] and its security considerations also apply to BGP sessions when carrying SR Policy information. The SR Policies distributed by BGP are expected to be used entirely within this trusted SR domain i.e. within a single AS or between multiple AS/domains within a single provider network. Therefore, precaution is necessary to ensure that the SR Policy information advertised via BGP sessions is limited to nodes in a secure manner within this trusted SR domain. BGP peering sessions for address-families other than SR Policy SAFI may be setup to routers outside the SR domain. The isolation of BGP SR Policy SAFI peering sessions may be used to ensure that the SR Policy information is not advertised by accident or error to an EBGP peering session outside the SR domain.

Additionally, it may be considered that the export of SR Policy information as described in this document constitutes a risk to confidentiality of mission-critical or commercially sensitive information about the network (more specifically endpoint/node addresses, SR SIDs and the SR Policies deployed). BGP peerings are not automatic and require configuration; thus, it is the responsibility of the network operator to ensure that only trusted nodes (that include both routers and controller applications) within the SR domain are configured to receive such information.

8. Acknowledgments

The authors of this document would like to thank Shyam Sethuram, John Scudder, Przemyslaw Krol, Alex Bogdanov, Nandan Saha, Bruno Decraene, Gurusiddesh Nidasesi, Kausik Majumdar, Zafar Ali, Swadesh Agarwal, Jakob Heitz, Viral Patel, Peng Shaofu and Cheng Li for their comments and review of this document.

9. Contributors

Arjun Sreekantiah
Cisco Systems
US

Email: asreekan@cisco.com

Acee Lindem
Cisco Systems
US

Email: acee@cisco.com

Siva Sivabalan
Cisco Systems
US

Email: msiva@cisco.com

Imtiyaz Mohammad
Arista Networks
India

Email: imtiyaz@arista.com

Gaurav Dawra
Cisco Systems
US

Email: gdawra.ietf@gmail.com

10. References

10.1. Normative References

[I-D.ietf-idr-tunnel-encaps]
Patel, K., Velde, G., Sangli, S., and J. Scudder, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-20 (work in progress), November 2020.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-24 (work in progress), October 2020.

[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>.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <https://www.rfc-editor.org/info/rfc3032>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, DOI 10.17487/RFC4360, February 2006, <https://www.rfc-editor.org/info/rfc4360>.

[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>.

[RFC7606] Chen, E., Ed., Scudder, J., Ed., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP UPDATE Messages", RFC 7606, DOI 10.17487/RFC7606, August 2015, <https://www.rfc-editor.org/info/rfc7606>.

[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>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8664] Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing", RFC 8664, DOI 10.17487/RFC8664, December 2019, <https://www.rfc-editor.org/info/rfc8664>.

10.2. Informational References

[RFC4272] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272, DOI 10.17487/RFC4272, January 2006, <https://www.rfc-editor.org/info/rfc4272>.

[RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <https://www.rfc-editor.org/info/rfc4456>.

[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>.

Appendix A. Deprecated Segment Sub-TLVs

This section describes the encoding of Segment Sub-TLVs that were specified in a previous version of this document and subsequently deprecated.

A.1. Type B-Deprecated: SID only, in the form of IPv6 address

The Type B-Deprecated Segment Sub-TLV encodes a single SRv6 SID. The format is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |  Length       |     Flags     |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                      SRv6 SID (16 octets)                     //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 2.

- Length is 18.

- Flags: 1 octet of flags as defined in Section 2.4.4.2.12.

- RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

- SRv6 SID: 16 octets of IPv6 address.

A.2. Type I-Deprecated: IPv6 Node Address with optional SRv6 SID

The Type I-Deprecated Segment Sub-TLV encodes an IPv6 node address, SR Algorithm and an optional SRv6 SID. The format is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Length      |     Flags     | SR Algorithm  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                 IPv6 Node Address (16 octets)               //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                    SRv6 SID (optional, 16 octets)           //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 10.

- Length is 34 when the SRv6 SID is present else is 18.

- Flags: 1 octet of flags as defined in Section 2.4.4.2.12.

- SR Algorithm: 1 octet specifying SR Algorithm as described in section 3.1.1 in [RFC8402], when A-Flag as defined in Section 2.4.4.2.12 is present. SR Algorithm is used by SRPM as described in section 4 in [I-D.ietf-spring-segment-routing-policy]. When A-Flag is not encoded, this field SHOULD be set to zero on transmission and MUST be ignored on receipt.

- IPv6 Node Address: a 16 octet IPv6 address.

- SRv6 SID: optional, 16 octet IPv6 address.

A.3. Type J-Deprecated: IPv6 Address + Interface ID for local and remote pair for SRv6 with optional SID

The Type J-Deprecated Segment Sub-TLV encodes an IPv6 Link Local adjacency with local node address, a local interface identifier (Local Interface ID), remote IPv6 node address, a remote interface identifier (Remote Interface ID) and an optional SRv6 SID. The format is as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |   Length      |     Flags     |   RESERVED    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                 Local Interface ID (4 octets)                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //                IPv6 Local Node Address (16 octets)          //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                 Remote Interface ID (4 octets)                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //                IPv6 Remote Node Address (16 octets)         //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //                SRv6 SID (optional, 16 octets)               //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 11.

- Length is 58 when the SRv6 SID is present else is 42.

- Flags: 1 octet of flags as defined in Section 2.4.4.2.12.

- RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

- Local Interface ID: 4 octets of interface index as defined in [RFC8664].

- IPv6 Local Node Address: a 16 octet IPv6 address.

- Remote Interface ID: 4 octets of interface index as defined in [RFC8664]. The value MAY be set to zero when the local node address and interface identifiers are sufficient to describe the link.

- IPv6 Remote Node Address: a 16 octet IPv6 address. The value MAY be set to zero when the local node address and interface identifiers are sufficient to describe the link.

- SRv6 SID: optional, 16 octet IPv6 address.

A.4. Type K-Deprecated: IPv6 Local and Remote addresses for SRv6 with optional SID

The Type K-Deprecated Segment Sub-TLV encodes an adjacency local address, an adjacency remote address and an optional SRv6 SID. The format is as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type       |   Length       |     Flags      |   RESERVED    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //               Local IPv6 Address (16 octets)                //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //               Remote IPv6 Address  (16 octets)             //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //                SRv6 SID (optional, 16 octets)              //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 12.
- Length is 50 when the SRv6 SID is present else is 34.
- Flags: 1 octet of flags as defined in Section 2.4.4.2.12.
- RESERVED: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.
- Local IPv6 Address: a 16 octet IPv6 address.
- Remote IPv6 Address: a 16 octet IPv6 address.
- SRv6 SID: optional, 16 octet IPv6 address.

Authors' Addresses

Stefano Previdi
Individual
IT

Email: stefano@previdi.net

Clarence Filsfils
Cisco Systems
Brussels
BE

Email: cfilsfil@cisco.com

Ketan Talaulikar (editor)
Cisco Systems
India

Email: ketant@cisco.com

Paul Mattes
Microsoft
One Microsoft Way
Redmond, WA 98052
USA

Email: pamattes@microsoft.com

Eric Rosen
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
US

Email: erosen@juniper.net

Dhanendra Jain
Google

Email: dhanendra.ietf@gmail.com

Steven Lin
Google

Email: stevenlin@google.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 6, 2021

W. J
Y. Liu
China Mobile
Y. Gu
Huawei

November 2, 2020

Traffic Steering using BGP Flowspec with SRv6 Policy
draft-jiang-idr-ts-flowspec-srv6-policy-01

Abstract

BGP Flow Specification (FlowSpec) [I-D.ietf-idr-rfc5575bis] has been proposed to distribute BGP FlowSpec NLRI to FlowSpec clients to mitigate (distributed) denial-of-service attacks, and to provide traffic filtering in the context of a BGP/MPLS VPN service. Recently, traffic steering applications in the context of SRv6 using FlowSpec aslo attract attention. This document introduces the usage of BGP FlowSpec to steer packets into an SRv6 Policy.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Segment Routing IPv6 (SRv6) is a protocol designed to forward IPv6 data packets on a network using the source routing model. SRv6 enables the ingress to add a segment routing header (SRH) [RFC8754] to an IPv6 packet and push an explicit IPv6 address stack into the SRH. After receiving the packet, each transit node updates the IPv6 destination IP address in the packet and segment list to implement hop-by-hop forwarding.

SRv6 Policy [I-D.ietf-spring-segment-routing-policy] is a tunneling technology developed based on SRv6. An SRv6 Policy is a set of candidate paths consisting of one or more segment lists, that is, segment ID (SID) lists. Each SID list identifies an end-to-end path from the source to the destination, instructing a device to forward traffic through the path rather than the shortest path computed using an IGP. The header of a packet steered into an SRv6 Policy is augmented with an ordered list of segments associated with that SRv6

Policy, so that other devices on the network can execute the instructions encapsulated into the list.

The headend of an SRv6 Policy may learn multiple candidate paths for an SRv6 Policy. Candidate paths may be learned via a number of different mechanisms, e.g., CLI, NetConf, PCEP, or BGP.

[I-D.ietf-idr-rfc5575bis] defines the flow specification (FlowSpec) that allows to convey flow specifications and traffic Action/Rules associated (rate- limiting, redirect, remark ...). BGP Flow specifications are encoded within the MP_REACH_NLRI and MP_UNREACH_NLRI attributes. Rules (Actions associated) are encoded in Extended Community attribute. The BGP Flow Specification function allows BGP Flow Specification routes that carry traffic policies to be transmitted to BGP Flow Specification peers to steer traffic.

This document proposes BGP flow specification usage that are used to steer data flow into an SRv6 Policy as well as to indicate Tailend function.

2. Definitions and Acronyms

- FlowSpec: Flow Specification
- SR: Segment Routing
- SRv6: IPv6 Segment Routing
- SID: Segment Identifier
- SRH: Segment Routing Header
- TE: Traffic Engineering

3. Operations

An SRv6 Policy [I-D.ietf-spring-segment-routing-policy] is identified through the tuple <headend, color, endpoint>. In the context of a specific headend, one may identify an SRv6 policy by the <color, endpoint> tuple. The headend is the node where the SRv6 policy is instantiated/implemented. The headend is specified as an IPv4 or IPv6 address and is expected to be unique in the domain. The endpoint indicates the destination of the SRv6 policy. The endpoint is specified as an IPv6 address and is expected to be unique in the domain. The color is a 32-bit numerical value that associates the SRv6 Policy, and it defines an application-level network Service Level Agreement (SLA) policy.

Assume one or multiple SRv6 Policies are already setup in the SRv6 HeadEnd device. In order to steer traffic into a specific SRv6 policy at the Headend, one can use the SRv6 color extended community and endpoint to map to a satisfying SRv6 policy, and steer traffic into this specific policy.

[I-D.ietf-idr-flowspec-redirect-ip] defines the redirect to IPv4 and IPv6 Next-hop action. The IPv6 next-hop address in the FlowSpec NLRI can be used to specify the endpoint of the SRv6 Policy. When the packets reach to the TailEnd device, some specific function imformation identifiers can be used decide how to further process the flows. Several endpoint functions are already defined, e.g., End.DT6: Endpoint with decapsulation and IPv6 table lookup, and End.DX6: Endpoint with decapsulation and IPv6 cross-connect. The BGP Prefix-SID defined in [RFC8669] is utilized to enable SRv6 VPN services [I-D.ietf-bess-srv6-services]. SRv6 Services TLVs within the BGP Prefix-SID Attribute can be used to indicate the endpoint functions.

This document proposes to carry the Color Extended Community and BGP Prefix-SID Attribute in the context of a Flowspec NLRI [I-D.ietf-idr-rfc5575bis] to an SRv6 Headend to steer traffic into one SRv6 policy, as well as to indicate specific Tailend functions.

In this document, the usage of at most one Color Extended Community in combination at most one BGP Prefix SID Attribute is discussed. For the case that a flowspec route carries multiple Color Extend Communities and/or a BGP Prefix SID Attribute, a protocol extension to Flowspec is required, and is thus out of the scope of this document.

However, the method proposed in this document still supports load balancing to the tailend device. To achieve that, the headend device CAN set up multiple paths in one SRv6 policy, and use a Flowspec route to indicate the specific SRv6 policy.

4. Application Example

In following scenario, BGP FlowSpec Controller signals the function imformation (SRv6 SID: Service_id_x) to the HeadEnd device.

```
      +------------+
      |   BGP FS   |
      | Controller |
      +------------+
         | Flowspec route to HeadEnd:
         |   NLRI: Filter Rules
         |   Redirect to IPv6 Nexthop: TailEnd's Address
         |   Policy Color: C1
         |   PrefixSID: Service_id_x
         |          .-----.
         |         (       )
         V     .--(         )--.
   +-------+  (                 )  +-------+
   |       |_( SRv6 Core Network )_|       |
   |HeadEnd| ( ================> ) |TailEnd|
   +-------+  (SR List<S1,S2,S3>)  +-------+
               '--(         )--'   Service SID: Service_id_x
                   (       )       (e.g.: End.DT4 or End.DT6 or others)
                    '_____'
```

Figure 1: Steering the Flow into SRv6 Policy

When the headend device (as a Flowspec client) receives such instructions, it will steer the flows matching the criteria in the Flowspec route into the SRv6 Policy matching the tuple (Endpoint: TailEnd's Address, Color: C1). And the packets of such flows will be encapsulated with SRH using the SR List<S1, S2, S3, Service_id_x>. When the packets reach to the TailEnd device, they will be further procetssed per the function denoted by the Service_id_x.

For the cases of intra-AS and inter-AS traffic steering using this method, the usages of Flowspec Color Extended Community with BGP prefix SID are the same for both scenarios. The difference lie between the local SRv6 policy configurations. For the inter-domain case, the operator can configure an inter-domain SRv6 policy/path at the Headend device. For the intra-domain case, the operator can configure an intra-domain SRv6 policy/path at the Headend device.

5. IANA Considerations

No IANA actions are required for this document.

6. Security Considerations

This document does not change the security properties of SRv6 and BGP.

7. Contributors

The following people made significant contributions to this document:

Shunwan Zhaung
Huawei
Email: zhuangshunwan@huawei.com

Haibo Wang
Huawei
Email: rainsword.wang@huawei.com

Jie Dong
Huawei
Email: jie.dong@huawei.com

8. Acknowledgements

TBD

9. References

9.1. Normative References

[I-D.ietf-bess-srv6-services]
Dawra, G., Filsfils, C., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "SRv6 BGP based Overlay services", draft-ietf-bess-srv6-services-04 (work in progress), July 2020.

[I-D.ietf-idr-flowspec-redirect-ip]
Uttaro, J., Haas, J., Texier, M., Andy, A., Ray, S., Simpson, A., and W. Henderickx, "BGP Flow-Spec Redirect to IP Action", draft-ietf-idr-flowspec-redirect-ip-02 (work in progress), February 2015.

[I-D.ietf-idr-rfc5575bis]
Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", draft-ietf-idr-rfc5575bis-26 (work in progress), August 2020.

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-09 (work in progress), May 2020.

[I-D.ietf-idr-tunnel-encaps]
Patel, K., Velde, G., Sangli, S., and J. Scudder, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-19 (work in progress), September 2020.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-08 (work in progress), July 2020.

[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>.

[RFC8669] Previdi, S., Filsfils, C., Lindem, A., Ed., Sreekantiah, A., and H. Gredler, "Segment Routing Prefix Segment Identifier Extensions for BGP", RFC 8669, DOI 10.17487/RFC8669, December 2019, <https://www.rfc-editor.org/info/rfc8669>.

9.2. Informative References

[RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <https://www.rfc-editor.org/info/rfc4456>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

Authors' Addresses

Wenying Jiang
China Mobile
Beijing
China

Email: jiangwenying@chinamobile.com

Yisong Liu
China Mobile
Beijing
China

Email: liuyisong@chinamobile.com

Yunan Gu
Huawei
Beijing
China

Email: guyunan@huawei.com

IDR WG
Internet-Draft
Intended status: Standards Track
Expires: April 17, 2021

Yao. Liu
Shaofu. Peng
ZTE Corporation

October 14, 2020

BGP extensions of SR policy for path protection
draft-lp-idr-sr-path-protection-00

Abstract

This document proposes extensions of BGP in order to provide path protection within a candidate path when delivering SR policy. And it also extends BGP-LS to provide some extra information of the segment list in a candidate path in the advertisement.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Segment Routing [RFC8402] allows a headend node to steer a packet flow along any path.

[I-D.ietf-spring-segment-routing-policy] details the concepts of SR Policy and steering into an SR Policy. An SR Policy is a set of candidate paths, each consisting of one or more segment lists. The headend of an SR Policy may learn multiple candidate paths for an SR Policy.

When SR policies are involved and a strict compliance of the policy is required, an end-to-end protection should be preferred over a local repair mechanism.

Candidate path can be used for path protection, that is, the lower preference candidate path may be designated as the backup for a specific or all (active) candidate path(s).

Despite using candidate path for protection, protection at segment list level within an candidate path is also useful.

[I-D.ietf-idr-segment-routing-te-policy] specifies BGP extensions for the advertisement of SR Policies and each candidate path is carried in an NLRI, if candidate path is used for protection ,the BGP origin must generate multiple NLRIs. If protection at segment list level is possible, only one NLRI is needed which makes the advertisement more simple.

On the other hand, protection in and between candidate path can be combined together to provide a more comprehensive protection mechanism.

This document proposes extensions of BGP in order to provide path protection in an candidate path when delivering SR policy.

[I-D.ietf-idr-te-lsp-distribution] describes a mechanism to collect the SR policy information that is locally available in a node and advertise it into BGP Link State (BGP-LS) updates. This document also extends it to provide some extra information of the segment list in a candidate path in the BGP-LS advertisement.

2. BGP Extensions for Advertising Segment List

2.1. Extensions of Segment List sub-TLV

Segment List sub-TLV is introduced in [I-D.ietf-idr-segment-routing-te-policy] and it includes the elements of the paths (i.e., segments).

This document introduces a one-bit flag in the RESERVED field.

```
 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             |B|  RESERVED   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                           sub-TLVs                          //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Segment List sub-TLV

B-Flag(Backup Flag): one bit. When set to 0, it indicates that the segment list acts as the active member in the candidate path. When set to 1, it indicates that the segment list acts as the backup path in the candidate path.

Using segment lists for path protection can be compatible with using candidate paths. When a path fails, the backup segment list within the same candidate path is used preferentially for path protection. If the backup list is also invalid, then other candidate path can be enabled for protection.

2.2. List Identifier Sub-TLV

This document introduces a new sub-sub-tlv of Segment List sub-TLV, where,

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |    Length     |            RESERVED           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      List Identifier                          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ~                      Optional TLVs                            ~
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: List Identifier Sub-TLV

Type: 1 octet. TBD.

Length: 1 octet, specifies the length of the value field not including Type and Length fields.

RESERVED: 2 octet of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.

List Identifier: 4 octets. It is the identifier of the corresponding segment list, so that the segment list can be operated according to the specified Segment List identifier.

This sub-TLV is optional and it MUST NOT appear more than once inside the Segment List sub-TLV.

2.2.1. List Protection Sub-TLV

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |    Length     |            RESERVED           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                    Backup  List ID 1                          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                           ...                                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                    Backup  List ID N                          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: List Identifier Sub-TLV

Type: 1 octet. TBD.

Length: 1 octet, specifies the length of the value field not including Type and Length fields.

RESERVED: 2 octet of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.

Backup List ID: 4 octets. It is the List Identifier of the backup segment list that protects this segment list. If there're multiple backup paths, the list ID of each path should be included in the TLV.

3. BGP-LS Extensions for Distributing Segment List States

[I-D.ietf-idr-te-lsp-distribution] describes a mechanism to collect the SR Policy information that is locally available in a node and advertise it into BGP Link State (BGP-LS) updates. The SR Policy information includes status of the candidate path, e.g, whether the candidate path is administrative shut or not.

SR Segment List TLV is defined in [I-D.ietf-idr-te-lsp-distribution] to to report the SID-List(s) of a candidate path. Figure 4 shows the flags in SR Segment List TLV.

```
        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|S|B|         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Flag Field of SR Segment List TLV

The meaning of the D,E,C,V,R,F,A,M can be found in [I-D.ietf-idr-te-lsp-distribution] section 6.5.

This document introduces two new flags, where,

S-Flag : Indicates the segment list is in administrative shut state when set.

B-Flag : Indicates the segment list is the backup path within the candidate path when set, otherwise it is the active path.

4. Security Considerations

Procedures and protocol extensions defined in this document do not affect the security considerations discussed in [I-D.ietf-idr-segment-routing-te-policy] and [I-D.ietf-idr-te-lsp-distribution].

5. IANA Considerations

TBD

6. Normative References

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-09 (work in progress), May 2020.

[I-D.ietf-idr-te-lsp-distribution]
Previdi, S., Talaulikar, K., Dong, J., Chen, M., Gredler, H., and J. Tantsura, "Distribution of Traffic Engineering (TE) Policies and State using BGP-LS", draft-ietf-idr-te-lsp-distribution-13 (work in progress), April 2020.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-08 (work in progress), July 2020.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

Authors' Addresses

Liu Yao
ZTE Corporation
Nanjing
China

Email: liu.yao71@zte.com.cn

Peng Shaofu
ZTE Corporation
Nanjing
China

Email: peng.shaofu@zte.com.cn

IDR
Internet-Draft
Intended status: Standards Track
Expires: April 5, 2021

F. Qin
China Mobile
H. Yuan
UnionPay
T. Zhou
G. Fioccola
Y. Wang
Huawei

October 2, 2020

BGP SR Policy Extensions to Enable IFIT
draft-qin-idr-sr-policy-ifit-04

Abstract

Segment Routing (SR) policy is a set of candidate SR paths consisting of one or more segment lists and necessary path attributes. It enables instantiation of an ordered list of segments with a specific intent for traffic steering. In-situ Flow Information Telemetry (IFIT) refers to network OAM data plane on-path telemetry techniques, in particular the most popular are In-situ OAM (IOAM) and Alternate Marking. This document defines extensions to BGP to distribute SR policies carrying IFIT information. So that IFIT methods can be enabled automatically when the SR policy is applied.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft will expire on April 5, 2021.

Copyright Notice

Table of Contents

1. Introduction

Segment Routing (SR) policy [I-D.ietf-spring-segment-routing-policy] is a set of candidate SR paths consisting of one or more segment lists and necessary path attributes. It enables instantiation of an ordered list of segments with a specific intent for traffic steering.

In-situ Flow Information Telemetry (IFIT) denotes a family of flow-oriented on-path telemetry techniques (e.g. IOAM, Alternate Marking), which can provide high-precision flow insight and real-time network issue notification (e.g., jitter, latency, packet loss).In particular, IFIT refers to network OAM data plane on-path telemetry techniques, including In-situ OAM (IOAM) [I-D.ietf-ippm-ioam-data] and Alternate Marking [RFC8321]. It can provide flow information on the entire forwarding path on a per-packet basis in real time.

An automatic network requires the Service Level Agreement (SLA) monitoring on the deployed service. So that the system can quickly detect the SLA violation or the performance degradation, hence to change the service deployment. For this reason, the SR policy native IFIT can facilitate the closed loop control and enable the automation of SR service.

This document defines extensions to Border Gateway Protocol (BGP) to distribute SR policies carrying IFIT information. So that IFIT behavior can be enabled automatically when the SR policy is applied.

This BGP extension allows to signal the IFIT capabilities together with the SR-policy. In this way IFIT methods are automatically activated and running. The flexibility and dynamicity of the IFIT applications are given by the use of additional functions on the controller and on the network nodes, but this is out of scope here.

2. Motivation

IFIT Methods are being introduced in multiple protocols and below is a proper picture of the relevant documents for Segment Routing. Indeed the IFIT methods are becoming mature for Segment Routing over the MPLS data plane (SR-MPLS) and Segment Routing over IPv6 data plane (SRv6), that is the main focus of this draft:

> IOAM: the reference documents for the data plane are [I-D.ietf-ippm-ioam-ipv6-options] for SRv6 and [I-D.gandhi-mpls-ioam-sr] for SR-MPLS.
>
> Alternate Marking: the reference documents for the data plane are [I-D.ietf-6man-ipv6-alt-mark] for SRv6 and [I-D.ietf-mpls-rfc6374-sfl], [I-D.gandhi-mpls-rfc6374-sr] for SR-MPLS.

The definition of these data plane IFIT methods for SR-MPLS and SRv6 imply requirements for various routing protocols, such as BGP, and this document aims to define BGP extensions to distribute SR policies carrying IFIT information. This allows to signal the IFIT capabilities so IFIT methods are automatically configured and ready to run when the SR Policy candidate paths are distributed through BGP.

It is to be noted that, for PCEP, [I-D.chen-pce-pcep-ifit] proposes the extensions to PCEP to distribute paths carrying IFIT information and therefore to enable IFIT methods for SR policy too.

3. IFIT methods for SR Policy

In-situ Operations, Administration, and Maintenance (IOAM) [I-D.ietf-ippm-ioam-data] records operational and telemetry information in the packet while the packet traverses a path between two points in the network. In terms of the classification given in RFC 7799 [RFC7799] IOAM could be categorized as Hybrid Type 1. IOAM mechanisms can be leveraged where active OAM do not apply or do not offer the desired results. When SR policy enables the IOAM, the IOAM header will be inserted into every packet of the traffic that is steered into the SR paths.

The Alternate Marking [RFC8321]technique is an hybrid performance measurement method, per RFC 7799 [RFC7799] classification of measurement methods. Because this method is based on marking consecutive batches of packets. It can be used to measure packet loss, latency, and jitter on live traffic.

This document aims to define the control plane. While the relevant documents for the data plane application of IOAM and Alternate Marking are respectively [I-D.ietf-ippm-ioam-ipv6-options] and [I-D.ietf-6man-ipv6-alt-mark] for Segment Routing over IPv6 data plane (SRv6).

4. IFIT Attributes in SR Policy

As defined in [I-D.ietf-idr-segment-routing-te-policy], the SR Policy encoding structure is as follows:

```
      SR Policy SAFI NLRI: <Distinguisher, Policy-Color, Endpoint>
      Attributes:
         Tunnel Encaps Attribute (23)
            Tunnel Type: SR Policy
                Binding SID
                Preference
                Priority
                Policy Name
                Explicit NULL Label Policy (ENLP)
                Segment List
                    Weight
                    Segment
                    Segment
                    ...
                ...
```

A candidate path includes multiple SR paths, each of which is specified by a segment list. IFIT can be applied to the candidate path, so that all the SR paths can be monitored in the same way. The new SR Policy encoding structure is expressed as below:

```
      SR Policy SAFI NLRI: <Distinguisher, Policy-Color, Endpoint>
      Attributes:
         Tunnel Encaps Attribute (23)
            Tunnel Type: SR Policy
                Binding SID
                Preference
                Priority
                Policy Name
                Explicit NULL Label Policy (ENLP)
                IFIT Attributes
                Segment List
                    Weight
                    Segment
                    Segment
                    ...
                ...
```

IFIT attributes can be attached at the candidate path level as sub-TLVs. There may be different IFIT tools. The following sections will describe the requirement and usage of different IFIT tools, and define the corresponding sub-TLV encoding in BGP.

Note that the IFIT attributes here described can also be generalized and included as sub-TLVs for other SAFIs and NLRIs.

5. IFIT Attributes Sub-TLV

The format of the IFIT Attributes Sub-TLV is defined 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
                                +---------------+---------------+
                                |      Type     |    Length     |
+-------------------------------+---------------+---------------+
|                                                               |
//                           sub-TLVs                          //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 1 IFIT Attributes Sub-TLV

Where:

Type: to be assigned by IANA.

Length: the total length of the value field not including Type and Length fields.

sub-TLVs currently defined:

* IOAM Pre-allocated Trace Option Sub-TLV,

* IOAM Incremental Trace Option Sub-TLV,

* IOAM Directly Export Option Sub-TLV,

* IOAM Edge-to-Edge Option Sub-TLV,

* Enhanced Alternate Marking (EAM) sub-TLV.

The presence of the IFIT Attributes Sub-TLV implies support of IFIT methods (IOAM and/or Alternate Marking). It is worth mentioning that IOAM and Alternate Marking can be activated one at a time or can coexist; so it is possible to have only IOAM or only Alternate Marking enabled as Sub-TLVs. The sub-TLVs currently defined for IOAM and Alternate Marking are detailed in the next sections.

5.1. IOAM Pre-allocated Trace Option Sub-TLV

The IOAM tracing data is expected to be collected at every node that a packet traverses to ensure visibility into the entire path a packet takes within an IOAM domain. The preallocated tracing option will create pre-allocated space for each node to populate its information.

The format of IOAM pre-allocated trace option sub-TLV is defined 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
   +---------------+---------------+-------------------------------+
   |    Type=1     |   Length=6    |    Namespace ID               |
   +---------------+---------------+--------------+--------+-------+
   |         IOAM Trace Type                      | Flags  | Rsvd  |
   +----------------------------------------------+--------+-------+
```

Fig. 2 IOAM Pre-allocated Trace Option Sub-TLV

Where:

Type: 1 (to be assigned by IANA).

Length: 6, it is the total length of the value field (not including Type and Length fields).

Namespace ID: A 16-bit identifier of an IOAM-Namespace. The definition is the same as described in section 4.4 of [I-D.ietf-ippm-ioam-data].

IOAM Trace Type: A 24-bit identifier which specifies which data types are used in the node data list. The definition is the same as described in section 4.4 of [I-D.ietf-ippm-ioam-data].

Flags: A 4-bit field. The definition is the same as described in [I-D.ietf-ippm-ioam-flags] and section 4.4 of [I-D.ietf-ippm-ioam-data].

Rsvd: A 4-bit field reserved for further usage. It MUST be zero.

5.2. IOAM Incremental Trace Option Sub-TLV

The incremental tracing option contains a variable node data fields where each node allocates and pushes its node data immediately following the option header.

The format of IOAM incremental trace option sub-TLV is defined 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
   +---------------+---------------+-------------------------------+
   |    Type=2     |   Length=6    |    Namespace ID               |
   +---------------+---------------+---------------+--------+------+
   |         IOAM Trace Type                       | Flags  | Rsvd |
   +-----------------------------------------------+--------+------+
```

Fig. 3 IOAM Incremental Trace Option Sub-TLV

Where:

Type: 2 (to be assigned by IANA).

Length: 6, it is the total length of the value field (not including Type and Length fields).

All the other fields definistion is the same as the pre-allocated trace option sub-TLV in section 4.1.

5.3. IOAM Directly Export Option Sub-TLV

IOAM directly export option is used as a trigger for IOAM data to be directly exported to a collector without being pushed into in-flight data packets.

The format of IOAM directly export option sub-TLV is defined 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
                                   +---------------+---------------+
                                   |    Type=3     |   Length=12   |
   +-----------------------------------------------+---------------+
   |        Namespace ID           |            Flags              |
   +-------------------------------+---------------+---------------+
   |               IOAM Trace Type                 |      Rsvd     |
   +-----------------------------------------------+---------------+
   |                         Flow ID                               |
   +---------------------------------------------------------------+
```

Fig. 4 IOAM Directly Export Option Sub-TLV

Where:

Type: 3 (to be assigned by IANA).

Length: 12, it is the total length of the value field (not including Type and Length fields).

Namespace ID: A 16-bit identifier of an IOAM-Namespace. The definition is the same as described in section 4.4 of [I-D.ietf-ippm-ioam-data].

IOAM Trace Type: A 24-bit identifier which specifies which data types are used in the node data list. The definition is the same as described in section 4.4 of [I-D.ietf-ippm-ioam-data].

Flags: A 16-bit field. The definition is the same as described in section 3.2 of [I-D.ietf-ippm-ioam-direct-export].

Flow ID: A 32-bit flow identifier. The definition is the same as described in section 3.2 of [I-D.ietf-ippm-ioam-direct-export].

Rsvd: A 4-bit field reserved for further usage. It MUST be zero.

5.4. IOAM Edge-to-Edge Option Sub-TLV

The IOAM edge to edge option is to carry data that is added by the IOAM encapsulating node and interpreted by IOAM decapsulating node.

The format of IOAM edge-to-edge option sub-TLV is defined 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
                                   +---------------+---------------+
                                   |    Type=4     |    Length=4   |
   +-------------------------------+---------------+---------------+
   |        Namespace ID           |         IOAM E2E Type         |
   +-------------------------------+-------------------------------+
```

Fig. 5 IOAM Edge-to-Edge Option Sub-TLV

Where:

Type: 4 (to be assigned by IANA).

Length: 4, it is the total length of the value field (not including Type and Length fields).

Namespace ID: A 16-bit identifier of an IOAM-Namespace. The definition is the same as described in section 4.6 of [I-D.ietf-ippm-ioam-data].

IOAM E2E Type: A 16-bit identifier which specifies which data types are used in the E2E option data. The definition is the same as described in section 4.6 of [I-D.ietf-ippm-ioam-data].

5.5. Enhanced Alternate Marking (EAM) sub-TLV

The format of Enhanced Alternate Marking (EAM) sub-TLV is defined 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
                                   +---------------+---------------+
                                   |    Type=5     |   Length=4    |
   +-------------------------------+-------+-------+-------+-------+
   |           FlowMonID                   |     Period    | Rsvd  |
   +---------------------------------------+---------------+-------+
```

Fig. 6 Enhanced Alternate Marking Sub-TLV

Where:

Type: 5 (to be assigned by IANA).

Length: 4, it is the total length of the value field (not including Type and Length fields).

FlowMonID: A 20-bit identifier to uniquely identify a monitored flow within the measurement domain. The definition is the same as described in section 5.3 of [I-D.ietf-6man-ipv6-alt-mark].

Period: Time interval between two alternate marking period. The unit is second.

Rsvd: A 4-bit field reserved for further usage. It MUST be zero.

6. SR Policy Operations with IFIT Attributes

The details of SR Policy installation and use are specified in [I-D.ietf-spring-segment-routing-policy]. This document complements SR Policy Operations described in [I-D.ietf-idr-segment-routing-te-policy] by adding the IFIT Attributes.

The operations described in [I-D.ietf-idr-segment-routing-te-policy] are always valid. The only difference is the addition of IFIT Attributes Sub-TLVs for the SR Policy NLRI, that can affect its acceptance by a BGP speaker, but the implementation MAY provide an option for ignoring the unrecognized or unsupported IFIT sub-TLVs. SR Policy NLRIs that have been determined acceptable, usable and valid can be evaluated for propagation, including the IFIT information.

The error handling actions are also described in [I-D.ietf-idr-segment-routing-te-policy].

The validation of the IFIT Attributes sub-TLVs introduced in this document MUST be performed to determine if they are malformed or invalid. The validation of the individual fields of the IFIT Attributes sub-TLVs are handled by the SRPM (SR Policy Module).

7. IANA Considerations

This document defines a new sub-TLV in the registry "BGP Tunnel Encapsulation Attribute sub-TLVs" to be assigned by IANA:

Codepoint	Description	Reference
TBD1	IFIT Attributes Sub-TLV	This document

This document requests creation of a new registry called "IFIT Attributes Sub-TLVs". The allocation policy of this registry is "Specification Required" according to RFC 8126 [RFC8126].

Following initial Sub-TLV codepoints are assigned by this document:

Value	Description	Reference
1	IOAM Pre-allocated Trace Option Sub-TLV	This document
2	IOAM Incremental Trace Option Sub-TLV	This document
3	IOAM Directly Export Option Sub-TLV	This document
4	IOAM Edge-to-Edge Option Sub-TLV	This document
5	Enhanced Alternate Marking Sub-TLV	This document

8. Security Considerations

The security mechanisms of the base BGP security model apply to the extensions described in this document as well. See the Security Considerations section of [I-D.ietf-idr-segment-routing-te-policy].

SR operates within a trusted SR domain RFC 8402 [RFC8402] and its security considerations also apply to BGP sessions when carrying SR Policy information. The isolation of BGP SR Policy SAFI peering sessions may be used to ensure that the SR Policy information is not advertised outside the SR domain. Additionally, only trusted nodes (that include both routers and controller applications) within the SR domain must be configured to receive such information.

Implementation of IFIT methods (IOAM and Alternate Marking) are mindful of security and privacy concerns, as explained in [I-D.ietf-ippm-ioam-data] and RFC 8321 [RFC8321]. Anyway incorrect IFIT parameters in the BGP extension SHOULD not have an adverse effect on the SR Policy as well as on the network, since it affects only the operation of the telemetry methodology.

9. Acknowledgements

The authors of this document would like to thank Ketan Talaulikar, Joel Halpern, Jie Dong for their comments and review of this document.

10. References

10.1. Normative References

[I-D.ietf-6man-ipv6-alt-mark]
Fioccola, G., Zhou, T., Cociglio, M., Qin, F., and R. Pang, "IPv6 Application of the Alternate Marking Method", draft-ietf-6man-ipv6-alt-mark-01 (work in progress), June 2020.

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-09 (work in progress), May 2020.

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-10 (work in progress), July 2020.

[I-D.ietf-ippm-ioam-direct-export]
Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ietf-ippm-ioam-direct-export-01 (work in progress), August 2020.

[I-D.ietf-ippm-ioam-flags]
Mizrahi, T., Brockners, F., Bhandari, S., Sivakolundu, R., Pignataro, C., Kfir, A., Gafni, B., Spiegel, M., and J. Lemon, "In-situ OAM Flags", draft-ietf-ippm-ioam-flags-02 (work in progress), July 2020.

[I-D.ietf-ippm-ioam-ipv6-options]
Bhandari, S., Brockners, F., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., Spiegel, M., Krishnan, S., Asati, R., and M. Smith, "In-situ OAM IPv6 Options", draft-ietf-ippm-ioam-ipv6-options-03 (work in progress), September 2020.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-08 (work in progress), July 2020.

[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>.

[RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <https://www.rfc-editor.org/info/rfc7799>.

[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>.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <https://www.rfc-editor.org/info/rfc8321>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

10.2. Informative References

[I-D.chen-pce-pcep-ifit]
Chen, H., Yuan, H., Zhou, T., Li, W., Fioccola, G., and Y. Wang, "Path Computation Element Communication Protocol (PCEP) Extensions to Enable IFIT", draft-chen-pce-pcep-ifit-01 (work in progress), September 2020.

[I-D.gandhi-mpls-ioam-sr]
Gandhi, R., Ali, Z., Filsfils, C., Brockners, F., Wen, B., and V. Kozak, "MPLS Data Plane Encapsulation for In-situ OAM Data", draft-gandhi-mpls-ioam-sr-03 (work in progress), September 2020.

[I-D.gandhi-mpls-rfc6374-sr]
Gandhi, R., Filsfils, C., Voyer, D., Salsano, S., and M. Chen, "Performance Measurement Using RFC 6374 for Segment Routing Networks with MPLS Data Plane", draft-gandhi-mpls-rfc6374-sr-05 (work in progress), June 2020.

[I-D.ietf-mpls-rfc6374-sfl]
Bryant, S., Swallow, G., Chen, M., Fioccola, G., and G. Mirsky, "RFC6374 Synonymous Flow Labels", draft-ietf-mpls-rfc6374-sfl-07 (work in progress), June 2020.

Appendix A.

Authors' Addresses

Fengwei Qin
China Mobile
No. 32 Xuanwumenxi Ave., Xicheng District
Beijing
China

Email: qinfengwei@chinamobile.com

Hang Yuan
UnionPay
1899 Gu-Tang Rd., Pudong
Shanghai
China

Email: yuanhang@unionpay.com

Tianran Zhou
Huawei
156 Beiqing Rd., Haidian District
Beijing
China

Email: zhoutianran@huawei.com

Giuseppe Fioccola
Huawei
Riesstrasse, 25
Munich
Germany

Email: giuseppe.fioccola@huawei.com

Yali Wang
Huawei
156 Beiqing Rd., Haidian District
Beijing
China

Email: wangyali11@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 1, 2021

Y. Wang
S. Zhuang
Y. Gu
Huawei

October 28, 2020

BGP Extension for Advertising In-situ Flow Information Telemetry (IFIT) Capabilities
draft-wang-idr-bgp-ifit-capabilities-01

Abstract

This document defines extensions to BGP to advertise the In-situ Flow Information Telemetry (IFIT) capabilities. Within an IFIT domain, IFIT-capability advertisement from the tail node to the head node assists the head node to determine whether a particular IFIT Option type can be encapsulated in data packets. Such advertisement would be useful for mitigating the leakage threat and facilitating the deployment of IFIT measurements on a per-service and on-demand basis.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft will expire on May 1, 2021.

Copyright Notice

Table of Contents

1. Introduction

In-situ Flow Information Telemetry (IFIT) denotes a family of flow-oriented on-path telemetry techniques, including In-situ OAM (IOAM) [I-D.ietf-ippm-ioam-data] and Alternate Marking [RFC8321], which can provide flow information on the entire forwarding path on a per-packet basis in real time (e.g., jitter, latency, packet loss).

The IFIT model describes how service flows are measured to obtain packet loss and latency. Specifically, IFIT measures the packet loss and latency of service flows on the ingress and egress of the transit network, and summarizes desired performance indicators. The IFIT model is composed of three objects: target flow, transit network, and measurement system. The transit network only bears target flows. The target flows are not generated or terminated on the transit network. The transit network can be a Layer 2 (L2), Layer 3 (L3), or L2+L3 hybrid network. Each node on the transit network must be reachable at the network layer. The measurement system consists of the ingress (configured with IFIT and IFIT parameters) and multiple IFIT-capable devices.

IFIT is a solution focusing on network domains. The "network domain" consists of a set of network devices or entities within a single administration. One network domain MAY consists of multiple IFIT domain. The family of emerging on-path flow telemetry techniques MAY be selectively or partially implemented in different vendors' devices as an emerging feature for various use cases of application-aware network operations, in addition, for some usecases, the IFIT Features are deployed on a per-service and on-demand basis. Within the IFIT domain, one or more IFIT-options are added into packet at the IFIT-enabled head node that is referred to as the IFIT encapsulating node. Then IFIT data fields MAY be updated by IFIT transit nodes that the packet traverses. Finally, the data fields are removed at a device that is referred to as the IFIT decapsulating node. Hence, a head node needs to know if the IFIT decapsulating node is able to support the IFIT capabilities.

This document defines extensions to BGP to advertise the IFIT capabilities of a tail node to a head node in an IFIT domain. Then the head node can learn the IFIT capabilities and determine whether a particular IFIT Option type can be encapsulated in traffic packets. Such advertisement would be useful for avoiding IFIT data leaking from the IFIT domain and facilitating the deployment of IFIT measurements on a per-service and on-demand basis.

2. Definitions and Acronyms

- IFIT: In-situ Flow Information Telemetry
- OAM: Operation Administration and Maintenance
- NLRI: Network Layer Reachable Information, the NLRI advertised in the BGP UPDATE as defined in [RFC4271] and [RFC4760] .

3. IFIT Capabilities

This document defines the IFIT Capabilities formed of a 16-bit bitmap. The following format is used:

```
                      0                   1
                      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
                     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                     |P|I|D|E|M|    Reserved         |
                     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1. IFIT Capabilities

- P-Flag: IOAM Pre-allocated Trace Option Type flag. When set, this indicates that the router is capable of IOAM Pre-allocated Trace [I-D.ietf-ippm-ioam-data].

- I-Flag: IOAM Incremental Trace Option Type flag. When set, this indicates that the router is capable of IOAM Incremental Tracing [I-D.ietf-ippm-ioam-data].

- D-Flag: IOAM DEX Option Type flag. When set, this indicates that the router is capable of IOAM DEX [I-D.ioamteam-ippm-ioam-direct-export].

- E-Flag: IOAM E2E Option Type flag. When set, this indicates that the router is capable of IOAM E2E processing [I-D.ietf-ippm-ioam-data].

- M-Flag: Alternate Marking flag. When set, this indicates that the router is capable of processing Alternative Marking packets [RFC8321].

- Reserved: Reserved for future use. They MUST be set to zero upon transmission and ignored upon receipt.

4. Option 1: Extension to BGP Extended Community for IFIT-Capability Advertisement

4.1. IPv4-Address-Specific IFIT Extended Community

For IPv4 networks, this section defines a new type of BGP extended community [RFC4360] called IPv4-Address-Specific IFIT Extended Community. The IPv4-Address-Specific IFIT Extended Community can be used by the IFIT decapsulation node to notify the IFIT Capabilities to its parterner (as the IFIT encapsulation node). It is a transitive extended community with type 0x01 and sub-type TBA.

The format of this extended community is shown in Figure 2.

```
     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 (0x01)  | Sub-Type (TBA)|     IFIT Capabilities         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               Originating IPv4 Address                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2. IPv4-Address-Specific IFIT Extended Community

- IFIT Capabilities: as defined in previous setion.

- Originating IPv4 Address field: A IPv4 address of the IFIT decapsulation node.

4.2. IPv6-Address-Specific IFIT Extended Community

For IPv6 networks, this section defines a new type of BGP extended community[RFC4360] called IPv6-Address-Specific IFIT Extended Community. The IPv6-Address-Specific IFIT Extended Community can be used by the IFIT decapsulation node to notify the IFIT Capabilities to its parterner (as the IFIT encapsulation node). It is a transitive IPv6 address specific extended community with type 0x00 and sub-type TBA.

The format of this extended community is shown in Figure 3.

```
     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 (0x00)  | Sub-Type (TBA)|    IFIT Capabilities          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               Originating IPv6 Address                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               Originating IPv6 Address (cont.)                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               Originating IPv6 Address (cont.)                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               Originating IPv6 Address (cont.)                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3. IPv6-Address-Specific IFIT Extended Community

- IFIT Capabilities: as defined in previous setion.

- Originating IPv6 Address field: A IPv6 address of the IFIT decapsulation node.

In this option, the Originating IP Address (inclue IPv4 and IPv6) in the extended community attribute is used as the IFIT decapsulation node.

5. Option 2: Extension to BGP Next-Hop Capability for IFIT-Capability Advertisement

The BGP Next-Hop Capability Attribute [I-D.ietf-idr-next-hop-capability] is a non-transitive BGP attribute, which is modified or deleted when the next-hop is changed, to reflect the capabilities of the new next-hop. The attribute consists of a set of Next-Hop Capabilities.

A Next-Hop Capability is a triple (Capability Code, Capability Length, Capability Value) aka a TLV:

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Capability Code (2 octets)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Capability Length (2 octets)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Capability Value (variable)   |
~                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4. BGP Next-Hop Capability

- o Capability Code: a two-octets unsigned binary integer which indicates the type of "Next-Hop Capability" advertised and unambiguously identifies an individual capability. This document defines a new Next-Hop Capability, which is called IFIT Next-Hop Capability. The Capability Code is TBD1.

- o Capability Length: a two-octets unsigned binary integer which indicates the length, in octets, of the Capability Value field. A length of 0 indicates that no Capability Value field is present.

- o Capability Value: a variable-length field. It is interpreted according to the value of the Capability Code. For the IFIT Next-Hop Capability, Capability Value contains IFIT Capabilities and Originate IP Address, as shown in the following figure.

```
                    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                    | IFIT Capabilities (2 octets)  |
                    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                    | Originate IP Address          |
                    | (4 or 16 octets)              |
                    ~                               ~
                    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5. BGP Capability Value for IFIT

- IFIT Capabilities: as defined in previous setion.

- Originate IP Address: An IPv4 or IPv6 Address of the IFIT decapsulation node.

A BGP speaker S that sends an UPDATE with the BGP Next-Hop Capability Attribute MAY include the IFIT Next-Hop Capability. The inclusion of the IFIT Next-Hop Capability with the NLRI advertised in the BGP UPDATE indicates that the BGP Next-Hop can act as the IFIT decapsulating node and it can process the specific IFIT encapsulation format indicated per the capability value. This is applied for all routes indicated in the same NRLI.

6. IANA Considerations

TBD

7. Security Considerations

This document defines extensions to BGP Extended Community and BGP Next-Hop Capability to advertise the IFIT capabilities. It does not introduce any new security risks to BGP.

8. Contributors

The following people made significant contributions to this document:

Weidong Li
Huawei
Email: poly.li@huawei.com

Haibo Wang
Huawei
Email: rainsword.wang@huawei.com

Tianran Zhou
Huawei
Email: zhoutianran@huawei.com

9. Acknowledgements

TBD

10. References

10.1. Normative References

[I-D.ietf-idr-next-hop-capability] Decraene, B., Kompella, K., and W. Henderickx, "BGP Next-Hop dependent capabilities", draft-ietf-idr-next-hop-capability-06 (work in progress), October 2020.

[I-D.ietf-ippm-ioam-data] Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-10 (work in progress), July 2020.

[I-D.ioamteam-ippm-ioam-direct-export] Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ioamteam-ippm-ioam-direct-export-00 (work in progress), October 2019.

[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>.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <https://www.rfc-editor.org/info/rfc8321>.

10.2. Informative References

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, DOI 10.17487/RFC4360, February 2006, <https://www.rfc-editor.org/info/rfc4360>.

[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>.

Authors' Addresses

Yali Wang
Huawei
Beijing
China

Email: wangyali11@huawei.com

Shunwan Zhuang
Huawei
Beijing
China

Email: zhuangshunwan@huawei.com

Yunan Gu
Huawei
Beijing
China

Email: guyunan@huawei.com

IDR Working Group W. Wang
Internet-Draft A. Wang
Intended status: Standards Track China Telecom
Expires: May 27, 2021 H. Wang
Huawei Technologies
G. Mishra
Verizon Inc.
S. Zhuang
J. Dong
Huawei Technologies
November 23, 2020

Route Distinguisher Outbound Route Filter (RD-ORF) for BGP-4
draft-wang-idr-rd-orf-05

Abstract

This draft defines a new Outbound Route Filter (ORF) type, called the Route Distinguisher ORF (RD-ORF). RD-ORF is applicable when the routers do not exchange VPN routing information directly (e.g. routers in single-domain connect via Route Reflector, or routers in Option B/Option AB/Option C cross-domain scenario).

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

With the rapid growth of network scale, Route Reflector is introduced in order to reduce the network complexity. Routers in the same Autonomous System only need to establish iBGP session with RR to transmit routes.

In VPN scenario shown in Figure 1, PE1 - PE4 establish IBGP sessions with RR to ensure the routes can be transmitted within AS100, where PE1 and PE3 maintain VRFs of VPN1 and VPN2, PE2 maintains VPN1's VRF and PE4 maintains VPN2's VRF. RR don not maintain any VRFs.

```
      +-----------------------------------------------+
      |                                               |
      |                                               |
      |   +---------+                   +---------+   |
      |   |   PE1   |                   |   PE4   |   |
      |   +---------+                   +---------+   |
      |    VPN1(RD1)  \               / VPN2(RD2)     |
      |    VPN2(RD2)    \+---------+ /                |
      |                  |         |                  |
      |                  |   RR    |                  |
      |                  |         |                  |
      |                  +---------+ \                |
      |                 /              \              |
      |   +---------+/                  +---------+   |
      |   |   PE2   |                   |   PE3   |   |
      |   +---------+                   +---------+   |
      |    VPN1(RD1)                     VPN1(RD1)    |
      |                     AS 100       VPN2(RD2)    |
      +-----------------------------------------------+
```

Figure 1: Single-domain scenario

When the VRF of VPN1 in PE1 overflows, due to PE1 and other PEs are not iBGP neighbors, BGP Maximum Prefix Features cannot work, so the problem on PE2 cannot be known.

Now, there are several solutions can be used to alleviate this problem:

- o Route Target Constraint (RTC) as defined in [RFC4684]

- o Address Prefix ORF as defined in [RFC5292]

- o PE-CE edge peer Maximum Prefix

- o Configure the Maximum Prefix for each VRF on edge nodes

However, there are limitations to existing solutions:

1) Route Target Constraint

RTC can only filter the VPN routes from the uninterested VRFs, if the "trashing routes" come from the interested VRF, filter on RTs will erase all prefixes from this VRF.

2) Address Prefix ORF

Using Address Prefix ORF to filter VPN routes need to pre-configuration, but it is impossible to know which prefix may cause overflow in advance.

3) PE-CE edge peer Maximum Prefix

This mechanism can only protect the edge between PE-CE, it can’t be deployed within PE that peered via RR. Depending solely on the edge protection is dangerous, because if only one of the edge points being comprised/error-configured/attacked, then all of PEs within domain are under risk.

4) Configure the Maximum Prefix for each VRF on edge nodes

When a VRF overflows, it stops the import of routes and log the extra VPN routes into its RIB. However, PEs should parse the BGP updates. These processes will cost CPU cycles and further burden the overflowing PE.

This draft defines a new ORF-type, called the Route Distinguisher ORF (RD-ORF). Using RD-ORF mechanism, VPN routes of a VPN can be controlled based on source RD. This mechanism is event-driven and does not need to be pre-configured. When a VRF of a router overflows, the router will find out the main source RD of VPN routes in this VRF, and send a RD-ORF to its BGP peer that carrys the RD. If a BGP speaker receives a RD-ORF from its BGP peer, it will filter the VPN routes it tends to send according to the RD-ORF entry.

RD-ORF is applicable when the routers do not exchange VPN routing information directly (e.g. routers in single-domain connect via Route Reflector, or routers in Option B/Option AB/Option C cross-domain scenario).

2. Conventions used in this document

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

3. Terminology

The following terms are defined in this draft:

- RD: Route Distinguisher, defined in [RFC4364]

- ORF: Outbound Route Filter, defined in [RFC5291]

- AFI: Address Family Identifier, defined in [RFC4760]

- SAFI: Subsequent Address Family Identifier, defined in [RFC4760]

- EVPN: BGP/MPLS Ethernet VPN, defined in [RFC7432]

- RR: Router Reflector, provides a simple solution to the problem of IBGP full mesh connection in large-scale IBGP implementation.

- VRF: Virtual Routing Forwarding, a virtual routing table based on VPN instance.

4. RD-ORF Encoding

In this draft, we defined a new ORF type called Route Distinguisher Outbound Route Filter (RD-ORF). The ORF entries are carried in the BGP ROUTE-REFRESH message as defined in [RFC5291]. A BGP ROUTE-REFRESH message can carry one or more ORF entries, and MUST be regenerated when it is tended to be sent to other BGP peers. The ROUTE-REFRESH message which carries ORF entries contains the following fields:

- AFI (2 octets)

- SAFI (1 octet)

- When-to-refresh (1 octet): the value is IMMEDIATE or DEFER

- ORF Type (1 octet)

- Length of ORF entries (2 octets)

A RD-ORF entry contains a common part and type-specific part. The common part is encoded as follows:

- Action (2 bits): the value is ADD, REMOVE or REMOVE-ALL

- Match (1 bit): the value is PERMIT or DENY

- Reserved (5 bits)

RD-ORF also contains type-specific part. The encoding of the type-specific part is shown in Figure 2.

```
             +-----------------------------------------+
             |                                         |
             |           Sequence (4 octets)           |
             |                                         |
             +-----------------------------------------+
             |                                         |
             |      Route Distinguisher (8 octets)     |
             |                                         |
             +-----------------------------------------+
```

Figure 2: RD-ORF type-specific encoding

- Sequence: identifying the order in which RD-ORF is generated

- Route Distinguisher: distinguish the different user routes. The RD-ORF filters the VPN routes it tends to send based on Route Distinguisher.

Note that if the Action component of an ORF entry specifies REMOVE-ALL, the ORF entry does not include the type-specific part.

When the BGP ROUTE-REFRESH message carries RD-ORF entries, it must be set as follows:

- The ORF-Type MUST be set to RD-ORF.

- The AFI MUST be set to IPv4, IPv6, or Layer 2 VPN (L2VPN).

- If the AFI is set to IPv4 or IPv6, the SAFI MUST be set to MPLS-labeled VPN address.

- If the AFI is set to L2VPN, the SAFI MUST be set to BGP EVPN.

- The Match field MUST be equal to DENY.

5. Application in single-domain scenario

5.1. Addition of RD-ORF entries

The operation of RD-ORF mechanism on each device is independent, each of them makes a local judgement to determine whether it needs to send RD-ORF to its peers.

In general, every VRF on PE is configured a Maximum Prefix, the trigger of RD-ORF mechanism can be set as the number of VPN routes in VRF reach 80% of the Maximum Prefix. For RR, it doesn’t have VRF and the machanism can be triggered by other conditions, such as the RR’s memory/CPU utilization reaches 80%.

When the RD-ORF mechanism is triggered, the device must send an alarm information to network operators.

5.1.1. Operation process of RD-ORF mechanism on source PE

In scenario shown in Figure 1, when the VRF of VPN1 in PE1 overflows, PE1 will do analysis and calculation locally to find out the main source of VPN routes in this VRF, assuming it is PE3. Then, PE1 will resolve the corresponding RD of VPN routes from BGP UPDATE message, and generate a BGP ROUTE-REFRESH message contains a RD-ORF entry, and send it to RR. The message contains the following fields:

- AFI is set to IPv4 , IPv6 or L2 VPN

- SAFI is set to "MPLS-labeled VPN address" or "BGP EVPN"

- When-to-refresh is set to IMMEDIATE

- ORF Type is set to RD-ORF

- Length of ORF entries depends on the type of Source Address sub-TLV (21, 23 or 33 octets)

- Action is set to ADD

- Match is set to DENY

- Sequence is set to 1

- Route Distinguisher is set to RD1

It noted that the Sequence can uniquely identifies an RD-ORF entry. All VRFs share the sequence field, and the corresponding sequence of RD-ORF sent by each VRF will be recorded on the device.

Sometimes, several VRFs in a PE may import VPN routes carries the same RT, as shown in Figure 3.

```
 +-----------------------------------------------+
 |      +--------+                               |
 |      |   RR   |                               |
 |      +--------+                               |
 |          |                                    |
 |          |                                    |
 |          |                                    |
 |          |                                    |
 |          |                                    |
 |      +--------+  VRF1 RT-import: RT1 RT3(RD3) |
 |      |   PE   |  VRF2 RT-import: RT2 RT3(RD3) |
 |      +--------+  VRF3 RT-import: RT3(RD3)     |
 |                  VRF4 RT-import: RT4          |
 |                                               |
 |                     AS 100                    |
 +-----------------------------------------------+
```

Figure 3: The scenario of several VRFs in a PE import VPN routes carries the same RT

In this scenario, VRF1, VRF2 and VRF3 import VPN routes carries RT3, which contains RD3. VRF1, VRF2 and VRF3 have different maximum prefix. When the VPN routes carrying RT3 cause the overflow of VRF3, PE will send a BGP ROUTE-REFRESH message containing a RD-ORF entry to RR, which Route Distinguisher field is equal to RD3. RR will stop sending associated VPN routes to PE. However, this will cause VRF1 to fail to receive VPN routes containing RD3.

The local determination of the PE can be used to inhibit the PE from sending RD-ORF entries. When the resources of the device are not exhausted, only prevent the overflowed VRF from importing related VPN routes without sending RD-ORF, unless all the VRFs that import the RD overflow.

5.1.2. Operation process of RD-ORF mechanism on RR

When RR receives the ROUTE-REFRESH message, it checks <AFI/SAFI, ORF-Type, Sequence, Route Distinguisher> to find whether it received the latest entry or not. If not, RR will discard the entry; otherwise, RR will add the RD-ORF entry into its Adj-RIB-out.

Before sending a VPN route toward PE1, RR will check its Adj-RIB-out and find there is a filter associated with RD1. Then, RR will stop sending that VPN route to PE1.

If the processing capacity of RR reaches the limit (e.g. RR’s memory/CPU utilization reaches 80%), RR will find out the peer that sends the most routing entries to it, assuming it is PE3. Then, RR will generate a BGP ROUTE-REFRESH message contains a RD-ORF entry based on the result of calculation, and send it to PE3.

5.1.3. Operation process of RD-ORF mechanism on target PE

After receiving the ROUTE-REFRESH message that carries a RD-ORF entry, PE3 will check if it receives the latest entry. If not, PE3 will discard it; otherwise, PE3 will add the RD-ORF entry into its Adj-RIB-out.

Before sending a VPN route toward RR, PE3 will check its Adj-RIB-out and find the RD-ORF entry prevent it from sending VPN route which carries RD1 to RR. Then, PE3 will stop sending that VPN route.

The BGP Maximum Prefix Features can be configured to protect PE-CE peering at the edge. Therefore, in general, CEs will not cause the overflow of PEs. If the boundary protection measures fail and cause the overflow, the PE can calculate and find the CEs in corresponding VRF, and break down the associated BGP sessions.

5.2. Withdraw of RD-ORF entries

When the RD-ORF mechanism is triggered, the alarm information will be generated and sent to the network operators. Operators should manually configure the network to resume normal operation. Due to devices can record the RD-ORF entries sent by each VRF, operators can find the entries needs to be withdrawn, and trigger the withdraw process as described in [RFC5291] manually. After returning to normal, the device sends withdraw ORF entries to its peers who have previously received ORF entries.

6. Applications in cross-domain scenarios

6.1. Application in Option B/Option AB cross-domain scenario

The Option B/Option AB cross-domain scenario is shown in Figure 4:

```
   +--------------------------+             +--------------------------+
   |                          |             |                          |
   |                          |             |                          |
   |   +---------+            |             |            +---------+   |
   |   |   PE1   |            |             |            |   PE3   |   |
   |   +---------+            |             |            +---------+   |
   |    VPN1(RD1) \           |             |           / VPN1(RD1)    |
   |    VPN2(RD2)   \+---------+    EBGP   +---------+/   VPN2(RD2)    |
   |                 |         |           |         |                 |
   |                 |  ASBR1  |-----------|  ASBR2  |                 |
   |                 |         |           |         |                 |
   |                 +---------+           +---------+                 |
   |                /         |             |         \                |
   |   +---------+/           |             |           \+---------+   |
   |   |   PE2   |            |             |            |   PE4   |   |
   |   +---------+            |             |            +---------+   |
   |    VPN1(RD1)             |             |             VPN2(RD2)    |
   |           AS1            |             |           AS2            |
   +--------------------------+             +--------------------------+
```

Figure 4: The Option B/Option AB cross-domain scenario

In Option B cross-domain scenario, PE1 - PE4 are responsible for maintaining VPN routing information in AS1 and AS2. There is a direct link between ASBR1 and ASBR2 via EBGP. In AS1, PE1 and PE2 establish IBGP sessions with ASBR1 to ensure the routes can be transmitted in AS1. In AS2, PE3 and PE4 establish IBGP session with ASBR2.

Due to the maintenance of VPN routes is only done by PEs. ASBRs cannot know whether the PEs' ability to handle VPN routes has reached the upper limit or not, so it needs the RD-ORF to control the number of routes.

Assume that PE1 - PE4 can transmit VPN routes through the network architecture shown in Figure 4. When the VRF of VPN1 in PE1 overflows, the RD-ORF mechanism will be implemented as follows:

1) PE1 will check and find out the main source of VPN routes in this VRF is PE3. Then, PE1 will resolve the corresponding RD from BGP UPDATE message, and generate a BGP ROUTE-REFRESH message contains an RD-ORF entry, and send it to ASBR1.

2) When ASBR1 receives the ROUTE-REFRESH message, it checks whether it receives the latest RD-ORF entry. If not, ASBR1 will discard the entry; Otherwise, ASBR1 will add the RD-ORF entry into its Adj-RIB-out.

Before sending a VPN route toward PE1, RR will check its Adj-RIB-out and find there is a filter associated with RD1. Then, ASBR1 will stop sending that VPN route.

Besides, ASBR1 will locally determine if it needs to send an RD-ORF entry to ASBR2. The judgment criteria refers to Section 5.1.2.

3) If ASBR2/PE3 receives the RD-ORF entry, it will repeat the above process.

When the RD-ORF mechanism is triggered, network operators need to manually configure the network to return to resume normal operation. The withdraw of RD-ORF entries refers to Section 5.2.

In Option AB cross-domain scenario, ASBRs maintain a VRF for a VPN. However, due to VPN routes in all VRFs use the same BGP session, ASBRs cannot prevent the overflow of a certain VRF by breaking down a BGP session. The operation process of RD-ORF is similar to that in Option B scenario.

6.2. Application in Option C cross-domain scenario

The Option C cross-domain scenario is shown in Figure 5:

```
                                    MP-EBGP
                +----------------------------------------+
                |                                        |
   +------------+------------+              +------------+------------+
   |       +----+----+       |              |       +----+----+       |
   |       |         |       |              |       |         |       |
   |  +----+   RR1   +----+  |              |  +----+   RR2   +----+  |
   |  |    |         |    |  |              |  |    |         |    |  |
   |  |    +---------+    |  |              |  |    +---------+    |  |
   |  |                   |  |              |  |                   |  |
   |  |IBGP           IBGP|  |              |  |IBGP           IBGP|  |
   |  |                   |  |              |  |                   |  |
 +-+--+----+         +----+--+-+          +-+--+----+         +----+--+-+
 |         |         |         |          |         |         |         |
 |   PE1   |         |  ASBR1  |----------|  ASBR2  |         |   PE2   |
 |         |         |         |          |         |         |         |
 +-+-------+   AS1   +-------+-+          +-+-------+   AS2   +-------+-+
   +-------------------------+              +-------------------------+
```

Figure 5: The Option C cross-domain scenario

In this scenario, PE1 and PE2 are responsible for maintaining VPN routing information in AS1 and AS2. In order to reduce the complexity that full-mesh brings to the network, RR1 and RR2 establish MP-EBGP session to transmit labeled routes. In AS1, PE1 and ASBR1 establish IBGP session with RR1 to ensure the routes can be transmitted in AS1. In AS2, PE2 and ASBR2 establish IBGP session with RR2.

Due to the maintenance of VPN routes is only done by PEs. RRs cannot know whether the PEs' ability to handle VPN routes has reached the upper limit or not, so it needs the RD-ORF to control the number of routes.

The operating mechanism of RD-ORF is similar to the description in Section 6.1.

7. Security Considerations

A BGP speaker will maintain the RD-ORF entries in Adj-RIB-out, this behavior consumes its memory and compute resources. To avoid the excessive consumption of resources, [RFC5291] specifies that a BGP speaker can only accept ORF entries transmitted by its interested peers.

8. IANA Considerations

This document defines a new Outbound Route Filter type - Route Distinguisher Outbound Route Filter (RD-ORF). The code point is from the "BGP Outbound Route Filtering (ORF) Types". It is recommended to set the code point of RD-ORF to 66.

9. Acknowledgement

Thanks Robert Raszuk, Jim Uttaro, Jakob Heitz, Jeff Tantsura, Rajiv Asati, John E Drake and Gert Doering for their valuable comments on this draft.

10. Normative References

[I-D.ietf-bess-evpn-inter-subnet-forwarding]
Sajassi, A., Salam, S., Thoria, S., Drake, J., and J. Rabadan, "Integrated Routing and Bridging in EVPN", draft-ietf-bess-evpn-inter-subnet-forwarding-11 (work in progress), October 2020.

[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>.

[RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, DOI 10.17487/RFC4360, February 2006, <https://www.rfc-editor.org/info/rfc4360>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[RFC4684] Marques, P., Bonica, R., Fang, L., Martini, L., Raszuk, R., Patel, K., and J. Guichard, "Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)", RFC 4684, DOI 10.17487/RFC4684, November 2006, <https://www.rfc-editor.org/info/rfc4684>.

[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>.

[RFC5291] Chen, E. and Y. Rekhter, "Outbound Route Filtering Capability for BGP-4", RFC 5291, DOI 10.17487/RFC5291, August 2008, <https://www.rfc-editor.org/info/rfc5291>.

[RFC5292] Chen, E. and S. Sangli, "Address-Prefix-Based Outbound Route Filter for BGP-4", RFC 5292, DOI 10.17487/RFC5292, August 2008, <https://www.rfc-editor.org/info/rfc5292>.

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <https://www.rfc-editor.org/info/rfc7432>.

Authors’ Addresses

Wei Wang
China Telecom
Beiqijia Town, Changping District
Beijing, Beijing 102209
China

Email: wangw36@chinatelecom.cn

Aijun Wang
China Telecom
Beiqijia Town, Changping District
Beijing, Beijing 102209
China

Email: wangaj3@chinatelecom.cn

Haibo Wang
Huawei Technologies
Huawei Building, No.156 Beiqing Rd.
Beijing, Beijing 100095
China

Email: rainsword.wang@huawei.com

Gyan S. Mishra
Verizon Inc.
13101 Columbia Pike
Silver Spring MD 20904
United States of America

Phone: 301 502-1347
Email: gyan.s.mishra@verizon.com

Shunwan Zhuang
Huawei Technologies
Huawei Building, No.156 Beiqing Rd.
Beijing, Beijing 100095
China

Email: zhuangshunwan@huawei.com

Jie Dong
Huawei Technologies
Huawei Building, No.156 Beiqing Rd.
Beijing, Beijing 100095
China

Email: jie.dong@huawei.com