IDR Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

J. Dong
Z. Hu
Z. Li
Huawei Technologies
X. Tang
R. Pang
China Unicom

February 22, 2021

BGP-LS Extensions for Segment Routing based Enhanced VPN
draft-dong-idr-bgpls-sr-enhanced-vpn-03

Abstract

Enhanced VPN (VPN+) aims to provide enhanced VPN services to support some applications' needs of enhanced isolation and stringent performance requirements. VPN+ requires integration between the overlay VPN connectivity and the characteristics provided by the underlay network. A Virtual Transport Network (VTN) is a virtual underlay network which consists of a customized network topology and a set of network resource allocated from the physical network. A VTN could be used as the underlay to support one or a group of VPN+ services.

This document specifies the BGP-LS mechanisms with necessary extensions to advertise the information of Segment Routing (SR) based VTNs to a centralized network controller. Each VTN can have a customized topology and a set of network resources allocated. Multiple VTNs may shared the same topology, and multiple VTNs may share the same set of network resources on some network segments. This allows flexible combination of network topology and network resource attributes to build a large number of VTNs with a relatively small number of logical topologies. The proposed mechanism is applicable to both segment routing with MPLS data plane (SR-MPLS) and segment routing with IPv6 data plane (SRv6).

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

Enhanced VPN (VPN+) is an enhancement to VPN services to support the needs of new applications, particularly the applications that are associated with 5G services. These applications require enhanced isolation and have more stringent performance requirements than that can be provided with traditional overlay VPNs. These properties require integration between the underlay and the overlay networks. [I-D.ietf-teas-enhanced-vpn] specifies the framework of enhanced VPN and describes the candidate component technologies in different network planes and layers. An enhanced VPN can be used for 5G network slicing, and will also be of use in more generic scenarios.

To meet the requirement of enhanced VPN services, a number of virtual underlay networks need to be created, each with a subset of the underlay network topology and a set of network resources allocated to meet the requirement of a specific VPN+ service or a group of VPN+ services. Such a virtual underlay network is called Virtual Transport Network (VTN) in [I-D.ietf-teas-enhanced-vpn].

[I-D.ietf-spring-resource-aware-segments] introduces resource-awareness to Segment Routing (SR) [RFC8402] by associating existing type of SIDs with network resource attributes (e.g. bandwidth, processing or storage resources). These resource-aware SIDs retain their original functionality, with the additional semantics of identifying the set of network resources available for the packet processing action. [I-D.ietf-spring-sr-for-enhanced-vpn] describes the use of resource-aware segments to build SR based VTNs. To allow the network controller and network nodes to perform VTN-specific explicit path computation and/or shortest path computation, the group of resource-aware SIDs allocated by network nodes to each VTN and the associated topology and resource attributes need to be distributed in the control plane.

When a VTN spans multiple IGP areas or multiple Autonomous Systems (ASes), BGP-LS is needed to advertise the VTN information in each IGP area or AS to the network controller, so that the controller could use the collected information to build the view of inter-area or inter-AS SR VTNs.

This document describes BGP-LS [RFC7752] based mechanism with necessary extensions to advertise the topology and resource attribute of inter-area and inter-domain SR based VTNs. Each VTN can have a customized topology and a set of network resources allocated. Multiple VTNs may shared the same topology, and multiple VTNs may share the same set of network resources on some network segments. This allows flexible combination of network topology and network resource attributes to build a large number of VTNs with a relatively small number of logical topologies. The definition of VTN is advertised as a node attribute using BGP-LS. The associated network topology and resources attributes of a VTN are advertised as link attributes using BGP-LS.

2. Advertisement of VTN Definition

According to [I-D.ietf-teas-enhanced-vpn], a VTN has a customized network topology and a set of dedicated or shared network resources. Thus a VTN can be defined as the combination of a set of network attributes, which include the topology attribute and other attributes, such as the associated network resources.

The Virtual Transport Network Definition (VTND) TLV is a new TLV of the optional BGP-LS Attribute which is associated with the node NLRI.

The format of VTND 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            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            VTN ID                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             MT-ID             |    Algorithm  |     Flags     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Sub-TLVs                             |
   ~                            ...                                ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

- Type: TBD

- Length: the length of the value field of the TLV. It is variable dependent on the included Sub-TLVs.

- VTN ID: A global significant 32-bit identifier which is used to identify a virtual transport network.

- o MT-ID: 16-bit identifier which indicates the multi-topology identifier of the IGP topology.

- o Algorithm: 8-bit identifier which indicates the algorithm which applies to this virtual transport network. It can be either a normal algorithm in [RFC8402] or a Flex-Algorithm [I-D.ietf-lsr-flex-algo].

- o Flags: 8-bit flags. Currently all the flags are reserved for future use. They SHOULD be set to zero on transmission and MUST be ignored on receipt.

- o Sub-TLVs: optional sub-TLVs to specify the additional attributes of a virtual transport network. Currently no sub-TLV is defined in this document.

3. Advertisement of VTN Topology Attribute

[I-D.dong-lsr-sr-enhanced-vpn] describes the IGP mechanisms to distribute the topology attributes of SR based VTNs. This section describes the BGP-LS mechanism to distribute both the intra-domain and inter-domain topology attributes of SR based VTNs.

3.1. Intra-domain Topology Advertisement

The intra-domain topology attribute of a VTN can be determined by the MT-ID and/or the algorithm ID included in the VTN definition. In practice, it could be described using two optional approaches.

The first approach is to use Multi-Topology Routing (MTR) [RFC4915] [RFC5120] with the segment routing extensions to advertise the topology associated with the SR based VTNs. Different algorithms MAY be used to further specify the computation algorithm or the metric type used for path computation within the topology. Multiple VTNs can be associated with the same <topology, algorithm>, and the IGP computation with the <topology, algorithm> tuple can be shared by these VTNs.

The second approach is to use Flex-Algo [I-D.ietf-lsr-flex-algo] to describe the topological constraints of SR based VTNs on a shared network topology (e.g. the default topology). Multiple VTNs can be associated with the same Flex-Algo, and the IGP computation with this Flex-Algo can be shared.

This section describes the two optional approaches to advertise the intra-domain topology of a VTN using BGP-LS.

3.1.1. MTR based Topology Advertisement

In section 4.2.2.1 of [I-D.ietf-idr-rfc7752bis], Multi-Topology Identifier (MT-ID) TLV is defined, which can contain one or more IS-IS or OSPF Multi-Topology IDs. The MT-ID TLV MAY be present in a Link Descriptor, a Prefix Descriptor, or the BGP-LS Attribute of a Node NLRI.

[I-D.ietf-idr-bgp-ls-segment-routing-ext] defines the BGP-LS extensions to carry the segment routing information using TLVs of BGP-LS Attribute. When MTR is used with SR-MPLS data plane, topology-specific prefix-SIDs and topology-specific Adj-SIDs can be carried in the BGP-LS Attribute associated with the prefix NLRI and link NLRI respectively, the MT-ID TLV is carried in the prefix descriptor or link descriptor to identify the corresponding topology of the SIDs.

[I-D.ietf-idr-bgpls-srv6-ext] defines the BGP-LS extensions to advertise SRv6 segments along with their functions and attributes. When MTR is used with SRv6 data plane, the SRv6 Locator TLV is carried in the BGP-LS Attribute associated with the prefix-NLRI, the MT-ID TLV can be carried in the prefix descriptor to identify the corresponding topology of the SRv6 Locator. The SRv6 End.X SIDs are carried in the BGP-LS Attribute associated with the link NLRI, the MT-ID TLV can be carried in the link descriptor to identify the corresponding topology of the End.X SIDs. The SRv6 SID NLRI is defined to advertise other types of SRv6 SIDs, in which the SRv6 SID Descriptors can include the MT-ID TLV so as to advertise topology-specific SRv6 SIDs.

[I-D.ietf-idr-rfc7752bis] also defines the rules of the usage of MT-ID TLV:

"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."

Editor’s note: the above rules indicates that only one MT-ID is allowed to be carried the Link or Prefix descriptors. When a link or prefix needs to be advertised in multiple topologies, multiple NLRIs needs to be generated to report all the topologies the link or prefix participates in, together with the topology-specific segment routing information and link attributes. This may increase the number of BGP Updates needed for advertising MT-specific topology attributes, and may introduce additional processing burden to both the sending BGP speaker and the receiving network controller. When the number of topologies in a network is not a small number, some optimization may be needed for the reporting of multi-topology information and the associated segment routing information in BGP-LS. Based on the WG's opinion, this will be elaborated in a future version.

3.1.2. Flex-Algo based Topology Advertisement

The Flex-Algo definition [I-D.ietf-lsr-flex-algo] can be used to describe the calculation-type, the metric-type and the topological constraints for path computation on a network topology. As specified in [I-D.dong-lsr-sr-enhanced-vpn], the topology of a VTN can be determined by applying Flex-Algo constraints on a particular topology.

BGP-LS extensions for Flex-Algo [I-D.ietf-idr-bgp-ls-flex-algo] provide the mechanisms to advertise the Flex-Algo definition information. BGP-LS extensions for SR-MPLS [I-D.ietf-idr-bgp-ls-segment-routing-ext] and SRv6 [I-D.ietf-idr-bgpls-srv6-ext] provide the mechanism to advertise the algorithm-specific segment routing information.

In[I-D.ietf-idr-bgp-ls-segment-routing-ext], algorithm-specific prefix-SIDs can be advertised in BGP-LS attribute associated with Prefix NLRI. In [I-D.ietf-idr-bgpls-srv6-ext], algorithm-specific SRv6 Locators can be advertised in BGP-LS Attribute associated with the corresponding Prefix NLRI, and algorithm-specific End.X SID can be advertised in BGP-LS Attribute associated with the corresponding Link NLRI. Other types of SRv6 SIDs can also be algorithm-specific and are advertised using the SRv6 SID NLRI.

3.2. Inter-Domain Topology Advertisement

In some network scenarios, a VTNs which span multiple areas or ASes needs to be created. The multi-domain VTN could have different inter-domain connectivity, and may be associated with different set of network resources in each domain and also on the inter-domain links. In order to build the multi-domain VTNs using segment routing, it is necessary to advertise the topology and resource attribute of VTN on the inter-domain links and the associated BGP Peering SIDs.

[I-D.ietf-idr-bgpls-segment-routing-epe] and [I-D.ietf-idr-bgpls-srv6-ext] defines the BGP-LS extensions for advertisement of BGP topology information between ASes and the associated BGP Peering Segment Identifiers. Such information could be used by a network controller for the computation and instantiation of inter-AS traffic engineering SR paths.

Depending on the network scenarios and the requirement of inter-domain VTNs, different mechanisms can be used to specify the inter-domain connections of VTNs.

- One EBGP session between two ASes can be established over multiple underlying links. In this case, different underlying links can be used for different inter-domain VTNs which requires link isolation between each other. In another similar case, the EBGP session is established over a single link, while the network resource (e.g. bandwidth) on this link can be partitioned into several pieces, each of which can be considered as a virtual member link. In both cases, different BGP Peer-Adj-SIDs SHOULD be allocated to each underlying physical or virtual member link, and ASBRs SHOULD advertise the VTN identifier associated with each BGP Peer-Adj-SID.

- For inter-domain connection between two ASes, multiple EBGP sessions can be established between different set of peering ASBRs. It is possible that some of these BGP sessions are used for one multi-domain VTN, while some other BGP sessions are used for another multi-domain VTN. In this case, different BGP peer-node-SIDs are allocated to each BGP session, and ASBRs SHOULD advertise the VTN identifier associated with each BGP Peer-node-SIDs.

- At the AS-level topology, different multi-domain VTNs may have different inter-domain connectivity. Different BGP Peer-Set-SIDs can be allocated to represent the groups of BGP peers which can be used for load-balancing in each multi-domain VTN.

In network scenarios where the MT-ID or Flex-Algo is used consistently in multiple areas or ASes covered by a VTN. the approaches to advertise topology-specific BGP peering SIDs are described as below:

- Using MT-based mechanism, the topology-specific BGP peering SIDs can be advertised with the MT-ID associated with the VTN carried in the corresponding link NLRI. This can be supported with the existing mechanisms defined in [RFC7752][I-D.ietf-idr-bgpls-segment-routing-epe] and [I-D.ietf-idr-bgpls-srv6-ext].

- Using Flex-Algo based mechanism, the topology-specific BGP peering SIDs can be advertised together with the Admin Group (color) of the corresponding Flex-Algo in the BGP-LS attribute.

In network scenarios where consistent usage of MT-ID or Flex-Algo among multiple ASes can not be expected, then the global-significant

VTN-ID can be used to define the AS level topologies. Within each domain, the MT or Flex-Algo based mechanism could still be used for topology advertisement.

3.2.1. VTN ID TLV

A new VTN ID TLV is defined to describe the identifiers of one or more VTNs an intra-domain or inter-domain link belongs to. It can be carried in BGP-LS attribute which is associated with a Link NLRI, or it could be carried as a sub-TLV in the L2 Bundle Member Attribute TLV.

The format of VTN ID TLV is as 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             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Flags            |           Reserved            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                            VTN ID-1                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ~                              ...                              ~
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                            VTN ID-n                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

- Type: TBD

- Length: The length of the value field of the sub-TLV. It is variable dependent on the number of VTN IDs included.

- Flags: 16 bit flags. All the bits are reserved, which MUST be set to 0 on transmission and ignored on receipt.

- Reserved: this field is reserved for future use. MUST be set to 0 on transmission and ignored on receipt.

- VTN IDs: One or more 32-bit identifiers to specify the VTNs this link or member link belongs to.

4. Advertisement of VTN Resource Attribute

[I-D.dong-lsr-sr-enhanced-vpn] specifies the mechanism to advertise the resource information associated with each VTN. It is based on the extensions to the advertisement of L2 bundle member links information[RFC8668]. This section defines the corresponding BGP-LS extensions. Two new TLVs are defined to carry the VTN ID and the link attribute flags of either a Layer-3 link or the L2 bundle member links. The VTN ID TLV is defined in section 3.2.1 of this document, and a new Link Attribute Flags TLV is defined in this section. The TE attributes of each Layer 3 link or the L2 bundle member link, such as the bandwidth and the SR SIDs, can be advertised using the mechanism as defined in [I-D.ietf-idr-bgp-ls-segment-routing-ext][I-D .ietf-idr-bgpls-segment-routing-epe] and [I-D.ietf-idr-bgpls-srv6-ext].

4.1. Link Attribute Flags TLV

A new Link attribute Flags TLV is defined to specify the characteristics of a link. It can be carried in BGP-LS attribute which is associated with a Link NLRI, or it could be carried as a sub-TLV in the L2 Bundle Member Attribute TLV. The format of the sub-TLV is as 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             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |            Flags              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

Type: TBD

Length: 4 octets.

Flags: 16-bit flags. This field is consistent with the Flag field in IS-IS Link Attribute sub-TLV in [RFC5029]. In addition to the flags defined in [RFC5029], A new Flag V is defined in this document. When the V flag is set, it indicates this link is a virtual link.

5. Advertisement of VTN specific Data Plane Identifiers

In network scenarios where each VTN is associated with an independent network topology or Flex-Algo, the topology or Flex-Algo specific SIDs or Locators could be used as the identifier of the VTN in data plane. In network scenarios where multiple VTNs share the same topology or Flex-Algo, additional data plane identifiers would be needed to identify different VTNs.

This section describes the mechanisms to advertise the VTN identifiers with different data plane encapsulations.

5.1. VTN-specific SR-MPLS SIDs

With SR-MPLS data plane, the VTN identification information is implicitly carried in the SR SIDs of the corresponding VTN. Each node SHOULD allocate VTN-specific Prefix-SIDs for each VTN it participates in. Similarly, VTN-specific Adj-SIDs MAY be allocated for each link which participates in the VTN.

5.1.1. VTN-specific Prefix-SID TLV

A new VTN-specific Prefix-SID TLV is defined to advertise the prefix-SID and its associated VTN. It is derived from VTN specific Prefix-SID sub-TLV of IS-IS [I-D.dong-lsr-sr-enhanced-vpn]. The format of the sub-TLV is as 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     |             Flags             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            VTN ID                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      SID/Index/Label(Variable)                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

- o Type: TBD

- o Length: The length of the value field of the sub-TLV. It is variable dependent on the length of the SID/Index/Label field.

- o Flags: 16-bit flags. The high-order 8 bits are the same as in the Prefix-SID sub-TLV defined in [RFC8667]. The lower-order 8 bits are reserved for future use, which SHOULD be set to 0 on transmission and MUST be ignored on receipt.

- o VTN ID: A 32-bit local identifier to identify the VTN this prefix-SID associates with.

- o SID/Index/Label: The same as defined in [RFC8667].

One or more of VTN-specific Prefix-SID TLVs MAY be carried in BGP-LS attribute of the associated Prefix NLRI. The MT-ID in the Prefix descriptors SHOULD be the same as the MT-ID in the definition of these VTNs.

5.1.2. VTN-specific Adj-SID TLV

A new VTN-specific Adj-SID TLV is defined to advertise the Adj-SID and its associated VTN. It is derived from VTN specific Adj-SID sub-TLV of IS-IS [I-D.dong-lsr-sr-enhanced-vpn]. The format of the sub-TLV is as 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     |            Flags              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                            VTN ID                             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      SID/Index/Label(Variable)                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

- o Type: TBD

- o Length: The length of the value field of the sub-TLV. It is variable dependent on the length of the SID/Index/Label field.

- o Flags: 16-bit flags. The high-order 8 bits are the same as in the Adj-SID sub-TLV defined in [RFC8667]. The lower-order 8 bits are reserved for future use, which SHOULD be set to 0 on transmission and MUST be ignored on receipt.

- o VTN ID: A 32-bit local identifier to identify the VTN this Adj-SID associates with.

- o SID/Index/Label: The same as defined in [RFC8667].

Multiple VTN-specific Adj-SID TLVs MAY be carried in BGP-LS attribute of the associated Link NLRI. The MT-ID in the Link descriptors SHOULD be the same as the MT-ID in the definition of these VTNs.

5.2. VTN-specific SRv6 Locators

With SRv6 data plane, the VTN identification information can be implicitly or explicitly carried in the SRv6 Locator of the corresponding VTN, this is to ensure that all network nodes (including both the SRv6 End nodes and Transit nodes) can identify the VTN to which a packet belongs to. Network nodes SHOULD allocate VTN-specific Locators for each VTN it participates in. The VTN-specific Locators are used as the covering prefix of VTN-specific SRv6 End SIDs, End.X SIDs and other types of SIDs.

Each VTN-specific SRv6 Locator MAY be advertised in a separate Prefix NLRI. If multiple VTNs share the same topology/algorithm, the topology/algorithm specific Locator is the covering prefix of a group of VTN-specific Locators. Then the advertisement of VTN-specific locators can be optimized to reduce the amount of information advertised in the control plane.

A new VTN locator-block sub-TLV under the SRv6 Locator TLV is defined to advertise a set of sub-blocks which follows the topology/algorithm specific Locator. Each VTN locator-block value is assigned to one of the VTNs which share the same topology/algorithm.

```
     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            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number of VTNs|  Block Length |            Reserved           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                            VTN ID #1                          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ~                       Locator Block Value                     ~
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ~                               ...                             ~
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                            VTN ID #n                          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ~                       Locator Block Value                     ~
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

o Type: TBD

o Length: The length of the value field of the sub-TLV. It is variable dependent on the number of VTNs and the Block Length.

- Number of VTNs: The number of VTNs which share the same topology/algorithm specific Locator as the covering prefix.

- Block Length: The length of the VTN locator-block which follows the length of the topology/algorithm specific Locator.

- VTN ID: A 32-bit identifier to identify the VTN the locator-block is associates with.

- Block Value: The value of the VTN locator-block for each VTN.

With the VTN locator-block sub-TLV, the VTN-specific Locator can be obtained by concatenating the topology/algorithm specific locator and the locator-block value advertised for the VTN.

5.3. Dedicated VTN ID in Data Plane

As the number of VTNs increases, with the mechanism described in [I-D.ietf-spring-sr-for-enhanced-vpn], the number of SR SIDs and SRv6 Locators allocated for different VTNs would also increase. In network scenarios where the number of SIDs or Locators becomes a concern, some data plane optimization may be needed to reduce the amount of SR SIDs and Locators allocated. As described in [I-D.dong-teas-enhanced-vpn-vtn-scalability], one approach is to decouple the data plane identifiers used for topology based forwarding and the identifiers used for the VTN-specific processing. Thus a dedicated data plane VTN-ID could be introduced and encapsulated in the packet. One possible encapsulation of VTN-ID in IPv6 data plane is proposed in [I-D.dong-6man-enhanced-vpn-vtn-id]. One possible encapsulation of VTN-ID in MPLS data plane is proposed in [I-D.li-mpls-enhanced-vpn-vtn-id].

In that case, the VTN ID encapsulated in data plane can be the same value as the VTN ID in the control plane, so that the overhead of advertising the mapping between the VTN IDs in the control plane and the corresponding data plane identifiers could be saved.

6. Security Considerations

This document introduces no additional security vulnerabilities to BGP-LS.

The mechanism proposed in this document is subject to the same vulnerabilities as any other protocol that relies on BGP-LS.

7. IANA Considerations

TBD

8. Acknowledgments

The authors would like to thank Shunwan Zhuang and Zhenbin Li for the review and discussion of this document.

9. References

9.1. Normative References

[I-D.ietf-idr-bgp-ls-flex-algo]
Talaulikar, K., Psenak, P., Zandi, S., and G. Dawra, "Flexible Algorithm Definition Advertisement with BGP Link-State", draft-ietf-idr-bgp-ls-flex-algo-05 (work in progress), November 2020.

[I-D.ietf-idr-bgp-ls-segment-routing-ext]
Previdi, S., Talaulikar, K., Filsfils, C., Gredler, H., and M. Chen, "BGP Link-State extensions for Segment Routing", draft-ietf-idr-bgp-ls-segment-routing-ext-16 (work in progress), June 2019.

[I-D.ietf-idr-bgpls-segment-routing-epe]
Previdi, S., Talaulikar, K., Filsfils, C., Patel, K., Ray, S., and J. Dong, "BGP-LS extensions for Segment Routing BGP Egress Peer Engineering", draft-ietf-idr-bgpls-segment-routing-epe-19 (work in progress), May 2019.

[I-D.ietf-idr-bgpls-srv6-ext]
Dawra, G., Filsfils, C., Talaulikar, K., Chen, M., daniel.bernier@bell.ca, d., and B. Decraene, "BGP Link State Extensions for SRv6", draft-ietf-idr-bgpls-srv6-ext-05 (work in progress), November 2020.

[I-D.ietf-idr-rfc7752bis]
Talaulikar, K., "Distribution of Link-State and Traffic Engineering Information Using BGP", draft-ietf-idr-rfc7752bis-05 (work in progress), November 2020.

[I-D.ietf-spring-resource-aware-segments]
Dong, J., Bryant, S., Miyasaka, T., Zhu, Y., Qin, F., Li, Z., and F. Clad, "Introducing Resource Awareness to SR Segments", draft-ietf-spring-resource-aware-segments-01 (work in progress), January 2021.

[I-D.ietf-spring-sr-for-enhanced-vpn]
Dong, J., Bryant, S., Miyasaka, T., Zhu, Y., Qin, F., Li, Z., and F. Clad, "Segment Routing based Virtual Transport Network (VTN) for Enhanced VPN", February 2021, <https://tools.ietf.org/html/draft-ietf-spring-sr-for-enhanced-vpn>.

[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>.

[RFC5029] Vasseur, JP. and S. Previdi, "Definition of an IS-IS Link Attribute Sub-TLV", RFC 5029, DOI 10.17487/RFC5029, September 2007, <https://www.rfc-editor.org/info/rfc5029>.

[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>.

[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>.

9.2. Informative References

[I-D.dong-6man-enhanced-vpn-vtn-id]
Dong, J., Li, Z., Xie, C., and C. Ma, "Carrying Virtual Transport Network Identifier in IPv6 Extension Header", draft-dong-6man-enhanced-vpn-vtn-id-02 (work in progress), November 2020.

[I-D.dong-lsr-sr-enhanced-vpn]
Dong, J., Hu, Z., Li, Z., Tang, X., Pang, R., JooHeon, L., and S. Bryant, "IGP Extensions for Segment Routing based Enhanced VPN", draft-dong-lsr-sr-enhanced-vpn-04 (work in progress), June 2020.

[I-D.dong-teas-enhanced-vpn-vtn-scalability]
Dong, J., Li, Z., Qin, F., and G. Yang, "Scalability Considerations for Enhanced VPN (VPN+)", draft-dong-teas-enhanced-vpn-vtn-scalability-01 (work in progress), November 2020.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-13 (work in progress), October 2020.

[I-D.ietf-lsr-isis-srv6-extensions]
Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extension to Support Segment Routing over IPv6 Dataplane", draft-ietf-lsr-isis-srv6-extensions-11 (work in progress), October 2020.

[I-D.ietf-teas-enhanced-vpn]
Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Networks (VPN+) Service", draft-ietf-teas-enhanced-vpn-06 (work in progress), July 2020.

[I-D.li-mpls-enhanced-vpn-vtn-id]
Li, Z. and J. Dong, "Carrying Virtual Transport Network Identifier in MPLS Packet", February 2021, <https://tools.ietf.org/html/draft-li-mpls-enhanced-vpn-vtn-id>.

[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>.

[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>.

[RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <https://www.rfc-editor.org/info/rfc8667>.

[RFC8668] Ginsberg, L., Ed., Bashandy, A., Filsfils, C., Nanduri, M., and E. Aries, "Advertising Layer 2 Bundle Member Link Attributes in IS-IS", RFC 8668, DOI 10.17487/RFC8668, December 2019, <https://www.rfc-editor.org/info/rfc8668>.

Authors' Addresses

Jie Dong
Huawei Technologies

Email: jie.dong@huawei.com

Zhibo Hu
Huawei Technologies

Email: huzhibo@huawei.com

Zhenbin Li
Huawei Technologies

Email: lizhenbin@huawei.com

Xiongyan Tang
China Unicom

Email: tangxy@chinaunicom.cn

Ran Pang
China Unicom

Email: pangran@chinaunicom.cn

BESS WorkGroup
Internet-Draft
Intended status: Standards Track
Expires: November 12, 2021

D. Rao
S. Agrawal
C. Filsfils
K. Talaulikar
Cisco Systems
D. Steinberg
Lapishills Consulting Limited
L. Jalil
Verizon
Y. Su
Alibaba, Inc
J. Guichard
Futurewei
K. Patel
Arrcus, Inc
H. Wang
Huawei Technologies

May 11, 2021

BGP Color-Aware Routing (CAR)
draft-dskc-bess-bgp-car-02

Abstract

This document describes a BGP based routing solution to establish end-to-end intent-aware paths across a multi-domain service provider transport network. This solution is called BGP Color-Aware Routing (BGP CAR).

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document specifies a new BGP SAFI called BGP Color-Aware Routing (BGP CAR). BGP CAR fulfills the transport and VPN problem statement and requirements described in [dskc-bess-bgp-car-problem-statement].

1.1. Terminology

Intent	Any combination of the following behaviors: a/ Topology path selection (e.g. minimize metric, avoid resource), b/ NFV service insertion (e.g. service chain steering), c/ per-hop behavior (e.g. 5G slice).
Color	A 32-bit numerical value associated with an intent: e.g. low-cost vs low-delay vs avoiding

	some resources.
Colored Service Route	An egress PE E2 colors its BGP VPN route V/v to indicate the intent that it requests for the traffic bound to V/v. The color is encoded as a BGP Color Extended community [I-D.ietf-idr-tunnel-encaps].
Color-Aware Path to (E2, C)	A routed path to E2 which satisfies the intent associated with color C. Several technologies may provide a Color-Aware Path to (E2, C): SR Policy [I-D.ietf-spring-segment-routing-policy], IGP Flex-Algo [I-D.ietf-lsr-flex-algo], BGP CAR [specified in this document].
Color-Aware Route (E2, C)	A distributed or signaled route that builds a color-aware path to E2 for color C.
Service Route Automated Steering on Color-aware path	E1 automatically steers a C-colored service route V/v from E2 onto an (E2, C) path. If several such paths exist, a preference scheme is used to select the best path: E.g. IGP Flex-Algo first then BGP CAR then SR Policy.
Color Domain	A set of nodes which share the same Color-to-Intent mapping. This set can be organized in one or several IGP instances or BGP domains.
Resolution of a BGP CAR route (E, C)	An inter-domain BGP CAR route (E, C) from N is resolved on an intra-domain color-aware path (N, C) where N is the next-hop of the BGP CAR route.
Resolution vs Steering	In this document and consistently with the terminology of the SR Policy document [I-D.ietf-spring-segment-routing-policy], steering is used to describe the mapping of a service route onto a BGP CAR path while the term resolution is preserved for the mapping of an inter-domain BGP CAR route on an intra-domain color-aware path. Service Steering: Service route -> BGP CAR path (or other Color-Aware Routed Paths: e.g., SR Policy) Intra-Domain Resolution: BGP CAR route -> intra-domain color aware path (e.g. SR Policy, IGP Flex-Algo, BGP CAR)

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

1.2. Illustration

Here is a brief illustration of the salient properties of the BGP CAR solution.

```
   +-------------+      +-------------+      +-------------+
   |             |      |             |      |             | V/v with C1
   |----+        |------|             |------|        +----|/
   | E1 |        |      |             |      |        | E2 |\
   |----+        |      |             |      |        +----| W/w with C2
   |             |------|             |------|             |
   |  Domain 1   |      |   Domain 2  |      |   Domain 3  |
   +-------------+      +-------------+      +-------------+
```

Figure 1

All the nodes are part of an interdomain network under a single authority and with a consistent color-to-intent mapping:

- C1 is mapped to "low-delay"
  - Flex-Algo FA1 is mapped to "low delay" and hence to C1
- C2 is mapped to "low-delay and avoid resource R"
  - Flex-Algo FA2 is mapped to "low delay and avoid resource R" and hence C2

E1 receives two service routes from E2:

- V/v with BGP Extended-Color community C1
- W/w with BGP Extended-Color community C2

E1 has the following color-aware paths:

- (E2, C1) provided by BGP CAR with the following per-domain support:
  - Domain1: over IGP FA1
  - Domain2: over SR Policy bound to color C1
  - Domain3: over IGP FA1

- (E2, C2) provided by SR Policy

E1 automatically steers the received service routes as follows:

- V/v via (E2, C1) provided by BGP CAR

- W/w via (E2, C2) provided by SR Policy

Illustrated Properties:

- Leverage of the BGP Color Extended-Community

  - The service routes are colored with widely-used BGP Extended-Color Community

- (E, C) Automated Steering

  - V/v and W/w are automatically steered on the appropriate color-aware path

- Seamless co-existence of BGP CAR and SR Policy

  - V/v is steered on BGP CAR color-aware path

  - W/w is steered on SR Policy color-aware path

- Seamless interworking of BGP CAR and SR Policy

  - V/v is steered on a BGP CAR color-aware path that is itself resolved within domain 2 onto an SR Policy bound to the color of V/v

Other properties:

- MPLS dataplane: with 300k PE's and 5 colors, the BGP CAR solution ensures that no single node needs to support a dataplane scaling in the order of Remote PE * C. This would otherwise blow the MPLS dataplane.

- Control-Plane: a node should not install a (E, C) path if it does not need it

- Incongruent Color-Intent mapping: the solution supports the signaling of a BGP CAR route across different color domains

The keys to this simplicity are:

- the leverage of the BGP Color Extended-Community to color service routes

- the definition of the automated steering: a C-colored service route V/v from E2 is steered onto a color-aware path (E2, C)

- the definition of the data model of a BGP CAR path: (E, C)

  * consistent with SR Policy data model

- the definition of the recursive resolution of a BGP CAR route: a BGP CAR (E2, C) via N is resolved onto the color-aware path (N, C) which may itself be provided by BGP CAR or via another color-aware routing solution: SR Policy, IGP Flex-Algo.

1.3. Requirements Language

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

2. BGP CAR SAFI

2.1. Data Model

The BGP CAR data model is:

- NLRI Key: IP Prefix, Color

- NLRI non-key encapsulation data: MPLS label stack, Label index, SRv6 SID list etc.

- BGP Next Hop

- AIGP Metric: accumulates color/intent specific metric across domains

- Local-Color-Mapping Extended-Community (LCM-EC): Optional 32-bit Color value used when a CAR route propagates between different color domains

2.2. Extensible encoding

Extensible encoding is ensured by:

- NLRI Route-Type field: provides extensibility to add new NLRI formats for new route-types

- Key length: field enables handling of unsupported route-types opaquely, enabling transitivity via RRs

- TLV-based encoding of non-key NLRI: enables support for multiple encapsulations with efficient update packing

- AIGP Attribute provides extensibility via TLVs, enabling definition of additional metric semantics for a color as needed for an intent

2.3. BGP CAR Route Origination

A BGP CAR route may be originated locally (e.g., loopback) or through redistribution of an (E, C) color-aware path provided by another routing solution: SR Policy, IGP Flex-Algo or BGP-LU [RFC8277].

2.4. BGP CAR Route Validation

A BGP CAR path (E, C) from N with encapsulation T is valid if color-aware path (N, C) exists and T is dataplane available.

A local policy may customize the validation process:

- the color constraint in the first check may be relaxed: instead N is reachable in the default routing table

- the dataplane availability constraint of T may be relaxed

- addition of a performance-measurement verification to ensure that the intent associated with C is met (e.g. delay < bound)

2.5. BGP CAR Route Resolution

A BGP color-aware route (E2, C1) from N is resolved over a color-aware route (N, C1). The color-aware route (N, C1) may be provided recursively by BGP CAR or by other routing solutions: SR Policy, IGP Flex-Algo, BGP-LU.

When multiple resolutions are possible, the default preference should be: IGP Flex-Algo, SR Policy, BGP CAR, BGP LU.

Through local policy, a BGP color-aware route (E2, C1) from N may be resolved over a color-aware route (N, C2): i.e. the local policy maps the resolution of C1 over C2. For example, in a domain where resource R is known to not be present, the inter-domain intent

C1="low delay and avoid R" may be resolved over an intra-domain path of intent C2="low delay".

The color-aware route (N, C1) may have a different dataplane encapsulation than the one of (E2, C1): e.g. a BGP CAR route (E2, C1) with SR-MPLS encapsulation may be transported over an intermediate SRv6 domain.

2.6. AIGP Metric Computation

The Accumulated IGP (AIGP) Attribute is updated as the BGP CAR route propagates across the network.

The value set (or appropriately incremented) in the AIGP TLV corresponds to the metric associated with the underlying intent of the color. For example, when the color is associated with a low-latency path, the metric value is set based on the delay metric.

Information regarding the metric type used by the underlying intra-domain mechanism can also be set.

If BGP CAR routes traverse across a discontinuity in the transport path for a given intent, add a penalty in accumulated IGP metric. The discontinuity is also indicated to upstream nodes via a bit in the AIGP TLV.

AIGP metric computation is recursive.

To avoid continuous IGP metric churn causing end to end BGP CAR churn, an implementation should provide thresholds to trigger AIGP update.

Additional AIGP extensions may be defined to signal state for specific use-cases: MSD along the BGP CAR advertisement, Minimum MTU along the BGP CAR advertisement.

2.7. Path Availability

The (E, C) route inherently provides availability of redundant paths at every hop. For instance, BGP CAR routes originated by two egress ABRs in a domain are advertised as multiple paths to ingress ABRs in the domain, where they become equal-cost or primary-backup paths. A failure of an egress ABR is detected and handled by ingress ABRs locally within the domain for faster convergence, without any necessity to propagate the event to upstream nodes for traffic restoration.

BGP ADD-PATH should be enabled for BGP CAR to signal multiple next hops through a transport RR.

2.8. BGP CAR signaling through different color domains

```
             [Color Domain 1   A]-----[B     Color Domain 2     E2]
             [C1=low-delay      ]     [C2=low-delay               ]
```

Let us assume a BGP CAR route (E2, C2) is signaled from B to A; two border routers of respectively domain 2 and domain 1. Let us assume that these two domains do not share the same color-to-intent mapping. Low-delay in domain 2 is color C2 while C1 in domain 1 (C1 <> C2).

The BGP CAR solution seamlessly supports this (rare) scenario while maintaining the separation and independence of the administrative authority in different color domains.

The solution works as follows:

- Within domain 2, the BGP CAR route is (E2, C2) via E2

- B signals to A the BGP CAR route as (E2, C2) via B with Local-Color-Mapping-Extended-Community (LCM-EC) of color C2

- A is aware (classic peering agreement) of the intent-to-color mapping within domain 2 ("low-delay" in domain 2 is C2)

- A maps C2 in LCM-EC to C1 and signals within domain 1 the received BGP CAR route as (E2, C2) via A with LCM-EC(C1)

- The nodes within the receiving domain 1 use the local color encoded in the LCM-EC for next-hop resolution and BGP CAR route installation

Salient properties:

- The NLRI never changes

- E is globally unique, which makes E-C in that order unique

- In the vast majority of the case, the color of the NLRI is used for resolution and steering

- In the rare case of color incongruence, the local color encoded in LCM-EC takes precedence

Further illustrations are provided in Appendix B.

2.9. Format and Encoding

BGP CAR leverages the BGP multi-protocol extensions [RFC4760] and uses the MP_REACH_NLRI and MP_UNREACH_NLRI attributes for route updates by using the SAFI value TBD1 along with AFI 1 for IPv4 prefixes and AFI 2 for IPv6 prefixes.

BGP speakers MUST use BGP Capabilities Advertisement to ensure support for processing of BGP CAR updates. This is done as specified in [RFC4760], by using capability code 1 (multi-protocol BGP), with AFI 1 and 2 (as required) and SAFI TBD1.

The sub-sections below specify the generic encoding of the BGP CAR NLRI followed by the encoding for specific NLRI types introduced in this document.

2.9.1. BGP CAR SAFI NLRI Format

The generic format for the BGP CAR SAFI NLRI is shown 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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  NLRI Length  |  Key Length   |   NLRI Type   |              //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+              //
   |                 Type-specific Key Fields                     //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          Type-specific Non-Key Fields (if applicable)        //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- o NLRI Length: 1 octet field that indicates the length in octets of the NLRI excluding the NLRI Length field itself.

- o Key Length: 1 octet field that indicates the length in octets of the NLRI type-specific key fields. Key length MUST be at least 2 less than the NLRI length.

- o NLRI Type: 1 octet field that indicates the type of the BGP CAR NLRI.

- o Type-Specific Key Fields: Depend on the NLRI type and of length indicated by the Key Length.

- o Type-Specific Non-Key Fields: optional and variable depending on the NLRI type. The NLRI encoding allows for encoding of specific non-key information associated with the route (i.e. the key) as part of the NLRI for efficient packing of BGP updates.

The indication of the key length enables BGP Speakers to determine the key portion of the NLRI and use it along with the NLRI Type field in an opaque manner for handling of unknown or unsupported NLRI types. This can help Route Reflectors (RR) to propagate NLRI types introduced in the future in a transparent manner.

The NLRI encoding allows for encoding of specific non-key information associated with the route (i.e. the key) as part of the NLRI for efficient packing of BGP updates.

The non-key portion of the NLRI MUST be omitted while carrying it within the MP_UNREACH_NLRI when withdrawing the route advertisement.

2.9.2. CAR NLRI Type

The Color-Aware Routes NLRI Type is used for advertisement of color-aware routes and has the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  NLRI Length  |  Key Length   |   NLRI Type   |Prefix Length  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               IP Prefix (variable)                           //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Color (4 octets)                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Followed by optional TLVs encoded as below:

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

where:

- NLRI Length: variable

- Key Length: variable

- NLRI Type: 1

- Type-Specific Key Fields: as below

- Prefix Length: 1 octet field that carries the length of prefix in bits. Length MUST be less than or equal to 32 for IPv4 (AFI=1) and less than or equal to 128 for IPv6 (AFI=2).

- IP Prefix: IPv4 or IPv6 prefix (based on the AFI). A variable size field that contains the most significant octets of the prefix, i.e., 1 octet for prefix length 1 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, and so on. The size of the field MUST be less than or equal to 4 for IPv4 (AFI=1) and less than or equal to 16 for IPv6 (AFI=2).

- Color: 4 octets that contains color value associated with the prefix.

o Type-Specific Non-Key Fields: specified in the form of optional TLVs as below:

- Type: 1 octet field that contains the type of the non-key TLV

- Length: 1 octet field that contains the length of the value portion of the non-key TLV in terms of octets

- Value: variable length field as indicated by the length field and to be interpreted as per the type field.

The prefix is routable across the administrative domain where BGP transport CAR is deployed. It is possible that the same prefix is originated by multiple BGP CAR speakers in the case of anycast addressing or multi-homing.

The Color is introduced to enable multiple route advertisements for the same prefix. The color is associated with an intent (e.g. low-latency) in originator color-domain.

The following sub-sections specify the non-key TLVs associated with the Color-Aware Routes NLRI type.

2.9.2.1. Label TLV

The Label TLV is used for advertisement of color-aware routes along with their MPLS labels and has the following format:

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

Followed by one (or more) Labels encoded as below:

```
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                 Label                 |Rsrv |S|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type : 1

- Length: variable, MUST be a multiple of 3

- Label Information: multiples of 3 octet fields to convey the MPLS label(s) associated with the advertised color-aware route. It is used for encoding a single label or a stack of labels as per procedures specified in [RFC8277].

When a BGP transport CAR speaker is propagating the route further after setting itself as the nexthop, it allocates a local label for the specific prefix and color combination which it updates in this TLV. It also MUST program a label cross-connect that would result in the label swap operation for the incoming label that it advertises with the label received from its best-path router(s).

2.9.2.2. Label Index TLV

The Label Index TLV is used for advertisement of Segment Routing MPLS (SR-MPLS) Segment Identifier (SID) [RFC8402] information associated with the labeled color-aware routes and has the following format:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |    Length     |    Reserved   |     Flags     ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~               |                 Label Index                   ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~               |
   +-+-+-+-+-+-+-+-+
```

where:

- Type : 2

- Length: 7

- Reserved: 1 octet field that MUST be set to 0 and ignored on receipt.

- Flags: 2 octet field that maps to the Flags field of the Label-Index TLV of the BGP Prefix SID Attribute [RFC8277].

- Label Index: 4 octet field that maps to the Label Index field of the Label-Index TLV of the BGP Prefix SID Attribute [RFC8277].

This TLV provides the equivalent functionality as Label-Index TLV of [RFC8669] for Transport CAR in SR-MPLS deployments. The BGP Prefix SID Attribute SHOULD be omitted from the labeled color-aware routes when the attribute is being used to only convey the Label Index TLV for better BGP packing efficiency.

When a BGP Transport CAR speaker is propagating the route further after setting itself as the nexthop, it allocates a local label for the specific prefix and color combination. When the received update has the Label Index TLV, it SHOULD use that hint to allocate the local label from the SR Global Block (SRGB) using procedures as specified in [RFC8669].

2.9.2.3. SRv6 SID TLV

BGP Transport CAR can be also used to setup end-to-end color-aware connectivity using Segment Routing over IPv6 (SRv6) [RFC8402]. [I-D.ietf-spring-srv6-network-programming] specifies the SRv6 Endpoint behaviors (e.g. End PSP) which MAY be leveraged for BGP CAR with SRv6.The SRv6 SID TLV is used for advertisement of color-aware routes along with their SRv6 SIDs and has the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |    Length     |  SRv6 SID Info (variable)   //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type : 3

- Length: variable, MUST be either less than or equal to 16, or be a multiple of 16

- SRv6 SID Information: field of size as indicated by the length that either carries the SRv6 SID(s) for the advertised color-aware route as one of the following:

  - A single 128-bit SRv6 SID or a stack of 128-bit SRv6 SIDs

  - A transposed portion (refer [I-D.ietf-bess-srv6-services]) of the SRv6 SID that MUST be of size in multiples of one octet and less than 16.

The BGP color-aware route update for SRv6 MUST include the BGP Prefix-SID attribute along with the TLV carrying the SRv6 SID information as specified in [I-D.ietf-bess-srv6-services] when using the transposition scheme of encoding for packing efficiency of BGP updates.

2.9.3. Local-Color-Mapping (LCM) Extended Community

This document defines a new BGP Extended Community called "LCM". The LCM is a Transitive Opaque Extended Community with the following encoding:

```
    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=0x3  | Sub-Type=TBD2 |          Reserved             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              Color                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- Type: 0x3

- Sub-Type: TBD2.

- Reserved: 2 octet of reserved field that MUST be set to zero on transmission and ignored on reception.

- Color: 4-octet field that carries the 32-bit color value.

When a CAR route crosses the originator color domain’s boundary, LCM EC is added. LCM EC conveys the local color mapping for the intent (e.g. low latency) into transit or remote color domains.

The LCM EC MAY be used for filtering of BGP CAR routes and/or for applying routing policies for the intent, when present.

2.10. Fault Handling

This the fault management actions as described in [RFC7606] are applicable for handling of BGP update messages for BGP-CAR.

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 BGP-CAR are being advertised over the same session. Alternately, the router MUST perform 'session reset' when the session is only being used for BGP-CAR.

3. Service route Automated Steering on Color-Aware path

E1 automatically steers a C-colored service route V/v from E2 onto an (E2, C) color-aware path. If several such paths exist, a preference scheme is used to select the best path: E.g. IGP Flex-Algo first then BGP CAR then SR Policy.

This is consistent with the automated service route steering on SR Policy (a routing solution providing color-aware path) defined in [I-D.ietf-spring-segment-routing-policy]. All the steering variations defined in [I-D.ietf-spring-segment-routing-policy] are applicable to BGP CAR color-aware path: on-demand steering, per-destination, per-flow, CO-only. For brevity, in this revision, we refer the reader to the [I-D.ietf-spring-segment-routing-policy] text.

Salient property: Seamless integration of BGP CAR and SR Policy.

Appendix A provides illustrations of service route automated steering.

4. Intents

The widely deployed color-aware path SR Policy solution demonstrates that the following intents can easily be associated with a color:

1. Minimization of a cost metric vs a latency metric

    * Minimization of different metric types, static and dynamic

2. Exclusion/Inclusion of SRLG and/or Link Affinity and/or minimum MTU/number of hops

3. Bandwidth management

4. In the inter-domain context, exclusion/inclusion of entire domains, and border routers

5. Inclusion of one or several virtual network function chains

   - Located in a regional domain and/or core domain, in a DC

6. Localization of the virtual network function chains

   - Some functions may be desired in the regional DC or vice versa

7. Per-Destination and Per-Flow steering

It is straightforward to note that the BGP CAR color-aware alternative supports intents 1, 2, 4 and 7.

Future revisions of this document will analyze the BGP CAR supports for 3, 5 and 6.

5. (E, C) Subscription and Filtering

This section defines an (E, C) BGP subscription model that allows to filter the (E, C) routes learned by a BGP CAR node.

5.1. Illustration

```
        E1-----------------A-------------------B-------------------E2
                                                <--- (E2, C1) ----
         -- F (E2, C1) -->   --- F (E2, C1) -->
                           |                   |
         <-- (E2, C1) ----   <--- (E2, C1) ----
```

- o BGP CAR route (E2, C1) advertised by E2 is not unconditionally distributed beyond a certain point (e.g., B)

- o E1 subscribes to (E2, C1) by advertising a filter route F (E2, C1) to its upstream peer A

- o If A has (E2, C1) in its BGP RIB, it will advertise (E2, C1) to E1

- o If A does not have (E2, C1), it will advertise F (E2, C1) to its peer B

- o B will advertise (E2, C1) to A, which will distribute it to E1

E1 may trigger a subscription for BGP CAR route (E2, C1) as a result of receiving a C1-colored service route V/v from E2, for on-demand steering via (E2, C1).

5.2. Definition

future version of this document

6. Scaling

This section analyses the key scale requirement of [ref:dskc-bess-bgp-car-problem-statement], specifically:

- No intermediate node dataplane should need to scale to (Colors * PEs)

- No node should learn and install a BGP CAR route to (E,C) if it does not install a Colored service route to E

Figure 2 provides an ultra-scale reference topology. Section 6.2 presents three design models to deploy BGP CAR in the reference topology. Section 6.3 analyses the scaling properties of each model. Section 6.4 illustrates the scaling benefits of the (E, C) BGP subscription and filtering.

6.1. Ultra-Scale Reference Topology

```
                                         RD:V/v via E2
          +-----+              +-----+ vpn label:30030 +-----+
  ....... |S-RR1| <........... |S-RR2| <...............|S-RR3| <......
  :       +-----+              +-----+  Color C1       +-----+      :
  :                                                                  :
  :                                                                  :
  :                                                                  :
 +:------------+--------------+--------------+--------------+--------:-+
 |:            |              |              |              |        : |
 |:            |              |              |              |        : |
 |:          +---+          +---+          +---+          +---+      : |
 |:          |121|          |231|          |341|          |451|      : |
 |:          +---+          +---+          +---+          +---+      : |
 |---+         |              |              |              |      +---|
 | E1|         |              |              |              |      | E2|
 |---+         |              |              |              |      +---|
 |           +---+          +---+          +---+          +---+        |
 |           |122|          |232|          |342|          |452|        |
 |           +---+          +---+          +---+          +---+        |
 |   Access    |   Metro      |   Core       |   Metro      | Access   |
 |   domain 1  |   domain 2   |   domain 3   |   domain 4   | domain 5 |
 +-------------+--------------+--------------+--------------+----------+
 iPE            iBRM          iBRC          eBRC          eBRM       ePE
```

Figure 2: Ultra-Scale Reference Topology

The following applies to the reference topology above:

- Independent ISIS/OSPF SR instance in each domain.

- Each domain has Flex Algo 128. Prefix SID for a node is SRGB 168000 plus node number.

- A BGP CAR route (E2, C1) is advertised by egress BRM node 451.The route is sourced locally from redistribution from IGP-FA 128.

- Not shown for simplicity, node 452 will also advertise (E2, C1).

- When a transport RR is used within the domain or across domains, ADD-PATH is enabled to advertise paths from both egress BRs to it’s clients.

- Egress PE E2 advertises a VPN route RD:V/v with BGP Color extended community C1 that propagates via service RRs to ingress PE E1.

- E1 steers V/v prefix via color-aware path (E2,C1) and VPN label 30030

6.2. Deployment model

6.2.1. Flat

```
                                         RD:V/v via E2
          +-----+              +-----+ vpn label:30030 +-----+
  ....... |S-RR1| <........... |S-RR2| <...............|S-RR3| <......
  :       +-----+              +-----+  Color C1       +-----+       :
  :                                                                  :
  :                                                                  :
  :                                                                  :
 +:------------+--------------+--------------+--------------+--------:-+
 |:            |              |              |              |        : |
 |:            |   (E2,C1)    |   (E2,C1)    |   (E2,C1)    |        : |
 |:          +---+ via 231  +---+ via 341  +---+ via 451  +---+      : |
 |:(E2,C1)   |121|<---------|231|<---------|341|<---------|451|      : |
 |: via 121 /+---+ L=168002 +---+ L=168002 +---+ L=168002 +---+      : |
 |---+     /   |              |              |              |      +---|
 | E1| <--/    |              |              |              |      | E2|
 |---+ L=168002|              |              |              |      +---|
 |           +---+          +---+          +---+          +---+        |
 |           |122|          |232|          |342|          |452|        |
 |           +---+          +---+          +---+          +---+        |
 |   Access    |   Metro      |   Core       |   Metro      | Access   |
 |   domain 1  |   domain 2   |   domain 3   |   domain 4   | domain 5 |
 +-------------+--------------+--------------+--------------+----------+
 iPE            iBRM          iBRC          eBRC          eBRM       ePE


 168121      168231        168341        168451
 168002      168002        168002        168002         168002
  30030       30030         30030         30030          30030     30030
```

Figure 3

1. Node 451 advertises BGP CAR route (E2, C1) to 341, from which it goes to 231 then to 121 and finally to E1

2. Each BGP hop allocates local label and programs swap entry in forwarding for (E2, C1)

3. E1 receives BGP CAR route (E2, C1) via 121 with label 168002

    1. Let’s assume E1 selects that path

4. E1 resolves BGP CAR route (E2, C1) via 121 on color-aware path (121, C1)

   1. Color-aware path (121, C1) is FA128 path to 121 (label 168121)

5. E1's imposition color-aware label-stack for V/v is thus

   1. 30030 <=> V/v

   2. 168002 <=> (E2, C1)

   3. 168121 <=> (121, C1)

6. Each BGP hop performs swap operation on 168002 bound to color-aware path (E2,C1)

6.2.2. Hierarchical Design with next-hop-self at ingress domain BR

```
                               (E2,C1)
                      +-----+  via 451        +-----+
                      |T-RR1| <-------------- |T-RR2|
                    / +-----+  L=168002       +-----+\
                   /                                   \
+-------------+---/----------+--------------+-----------\--+----------+
|             |  /           |              |            \ |          |
|  (E2,C1)    | / (451,C1)   |   (451,C1)   |             \|          |
|  via 121  +---+ via 231  +---+ via 341  +---+          +---+        |
|  L=168002 |121| <======= |231| <========|341| <======= |451|        |
|         / +---+ L=168451 +---+ L=168451 +---+          +---+        |
|---+    /    |              |              |              |      +---|
| E1|<--/     |              |              |              |      | E2|
|---+         |              |              |              |      +---|
|           +---+          +---+          +---+          +---+        |
|           |122|          |232|          |342|          |452|        |
|           +---+          +---+          +---+          +---+        |
|   Access    |   Metro      |   Core       |   Metro      | Access   |
|   domain 1  |   domain 2   |   domain 3   |   domain 4   | domain 5 |
+-------------+--------------+--------------+--------------+----------+
iPE            iBRM            iBRC          eBRC          eBRM      ePE

            168231        168341
168121      168451        168451        168451
168002      168002        168002        168002         168002
 30030       30030         30030         30030          30030     30030
```

Figure 4: Heirarchical BGP transport CAR, NHS at iBR

1. Node 451 advertises BGP CAR route (451, C1) to 341, from which it goes to 231 and finally to 121

2. Each BGP hop allocates local label and programs swap entry in forwarding for (451, C1)

3. 121 resolves received BGP CAR route (451, C1) via 231 (label 168451) on color-aware path (231, C1)

    1. Color-aware path (231, C1) is FA128 path to 231 (label 168231)

4. 451 advertises BGP CAR route (E2, C1) via 451 to Transport RR T-RR2, which reflects it to T-RR1, which reflects it to 121

5. 121 receives BGP CAR route (E2, C1) via 451 with label 168002

    1. Let’s assume 121 selects that path

6. 121 resolves BGP CAR route (E2, C1) via 451 on color-aware path (451, C1)

    1. Color-aware path (451, C1) is BGP CAR path to 451 (label 168451)

7. 121 imposition of color-aware label stack for (E2, C1) is thus

    1. 168002 <=> (E2, C1)

    2. 168451 <=> (451, C1)

    3. 168231 <=> (231, C1)

8. 121 advertises (E2, C1) to E1 with next hop self (121) and label 168002

9. E1 constructs same imposition color-aware label-stack for V/v via (E2, C1) as in the flat model:

    1. 30030 <=> V/v

    2. 168002 <=> (E2, C1)

    3. 168121 <=> (121, C1)

10. 121 performs swap operation on 168002 with hierarchical color-aware label stack for (E2, C1) via 451 from step 7

11. Nodes 231 and 341 perform swap operation on 168451 bound to color-aware path (451, C1)

12. 451 performs swap operation on 168002 bound to color-aware path (E2, C1)

Note: E1 does not need the BGP CAR (451, C1) route

6.2.3. Hierarchical Design with Next Hop Unchanged at ingress domain BR

```
                               (E2,C1)
                      +-----+  via 451        +-----+
                      |T-RR1| <-------------- |T-RR2|
                    / +-----+  L=168002       +-----+\
                   /                                   \
+-------------+---/----------+--------------+-----------\--+----------+
|             |  /           |              |            \ |          |
|  (E2,C1)    | / (451,C1)   |   (451,C1)   |             \|          |
|  via 451  +---+ via 231  +---+ via 341  +---+          +---+        |
|  L=168002/|121| <======= |231| <========|341| <======= |451|        |
|         / +---+ L=168451 +---+ L=168451 +---+          +---+        |
|---+ <--/  //|              |              |              |      +---|
| E1|      // |              |              |              |      | E2|
|---+ <===//  |              |              |              |      +---|
|  (451,C1) +---+          +---+          +---+          +---+        |
|  via 121  |122|          |232|          |342|          |452|        |
|  L=168451 +---+          +---+          +---+          +---+        |
|             |              |              |              |          |
|   Access    |   Metro      |   Core       |   Metro      | Access   |
|   domain 1  |   domain 2   |   domain 3   |   domain 4   | domain 5 |
+-------------+--------------+--------------+--------------+----------+
iPE            iBRM            iBRC          eBRC          eBRM      ePE

168121      168231        168341
168451      168451        168451        168451
168002      168002        168002        168002         168002
 30030       30030         30030         30030          30030     30030
```

Figure 5: Heirarchical BGP transport CAR, NHU at iBR

1. Nodes 341, 231 and 121 receive and resolve BGP CAR route (451, C1) the same as in the previous model

2. Node 121 allocates local label and programs swap entry in forwarding for (451, C1)

3. 451 advertises BGP CAR route (E2, C1) to Transport RR T-RR2, which reflects it to T-RR1, which reflects it to 121

4. Node 121 advertises (E2, C1) to E1 with next hop as 451 i.e. next-hop unchanged

5. 121 also advertises (451, C1) to E1 with next hop self (121) and label 168451

6. E1 resolves BGP CAR route (451, C1) via 121 on color-aware path (121, C1)

    1. Color-aware path (121, C1) is FA128 path to 121 (label 168121)

7. E1 receives BGP CAR route (E2, C1) via 451 with label 168002

    1. Let’s assume E1 selects that path

8. E1 resolves BGP CAR route (E2, C1) via 451 on color-aware path (451, C1)

    1. Color-aware path (451, C1) is BGP CAR path to 451 (label 168451)

9. E1’s imposition color-aware label-stack for V/v is thus

    1. 30030 <=> V/v

    2. 168002 <=> (E2, C1)

    3. 168451 <=> (451, C1)

    4. 168121 <=> (121, C1)

10. Nodes 121, 231 and 341 perform swap operation on 168451 bound to (451, C1)

11. 451 performs swap operation on 168002 bound to color-aware path (E2, C1)

6.3. Scale Analysis

The following two tables summarize the control-plane and dataplane scale of these three models:

	E1	121	231
FLAT	(E2,C) via (121,C)	(E2,C) via (231,C)	(E2,C) via (341,C)
H.NHS	(E2,C) via (121,C)	(E2,C) via (451,C) (451,C) via (231,C)	(451,C) via (341,C)
H.NHU	(E2,C) via (451,C) (451,C) via (121,C)	(451,C) via (231,C)	(451,C) via (341,C)

	E1	121	231
FLAT	V -> 30030 168002 168121	168002 -> 168002 168231	168002 -> 168002 168341
H.NHS	V -> 30030 168002 168121	168002 -> 168002 168451 168231	168451 -> 168451 168341
H.NHU	V -> 30030 168002 168451 168121	168451 -> 168451 168231	168451 -> 168451 168341

- The flat model is the simplest design, with a single BGP transport level. It results in the minimum label/SID stack at each BGP hop. However, it significantly increases the scale impact on the core BRs (e.g. 341), whose FIB capacity and even MPLS label space may be exceeded.

  * 341’s dataplane scales with (E2,C) where there may be 300k E’s and 5 C’s hence 1.5M entries > 1M MPLS dataplane

- The hierarchical models avoid the need for core BRs to learn routes and install label forwarding entries for (E, C) routes.

  * Whether NH self or unchanged at 121, 341’s dataplane scales with (451,C) where there may be thousands of 451’s and 5 C’s hence well under the 1M MPLS dataplane

- The next-hop-self option at ingress BRM (e.g. 121) hides the hierarchical design from the ingress PE, keeping its outgoing label programming as simple as the flat model. However, the ingress BRM requires an additional BGP transport level recursion, which coupled with load-balancing adds dataplane complexity. It needs to support a swap and push operation. It also needs to install label forwarding entries for the egress PEs that are of interest to its local ingress PEs.

- With the next-hop-unchanged option at ingress BRM (e.g. 121), only an ingress PE needs to learn and install output label entries for egress (E, C) routes. The ingress BRM only installs label forwarding entries for the egress ABR (e.g. 451). However, the ingress PE needs an additional BGP transport level recursion and pushes a BGP VPN label and two BGP transport labels. It may also need to handle load-balancing for the egress ABRs. This is the most complex dataplane option for the ingress PE.

6.4. Scaling Benefits of the (E, C) BGP Subscription and Filtering

The (E, C) subscription scheme from Section 5 provides the following scaling benefits for the models in Section 6.2

- An ingress PE (E1) only learns (E, C) routes that it needs to install into data plane for service route automated steering

- An ingress BRM (121) only learns (E, C) routes that it needs to install into data plane (for Next-Hop-Self), or that it needs to distribute towards it’s ingress PEs (inline RR with Next-Hop-Unchanged)

- An ingress BRM or a transport RR only needs to distribute the necessary subset of (E, C) routes to each client (subscriber); this minimizes their processing load for generating updates

- As a result, withdrawal of (E, C) routes when a remote node fails (E2), may also be faster, aiding better convergence

6.5. Anycast SID

This section describes how Anycast SID complements and improves the scaling designs above.

6.5.1. Anycast SID for transit inter-domain nodes

- Redundant BRs (e.g. two egress BRMs, 451 and 452) advertise BGP CAR routes for a local PE (e.g., E2) with the same SID (based on label-index). Such egress BRMs may be assigned a common Anycast SID, so that the BGP next-hops for these routes will also resolve via a color-aware path to the Anycast SID.

- The use of Anycast SID naturally provides fast local convergence upon failure of an egress BRM node. In addition, it decreases the recursive resolution and load-balancing complexity at an ingress BRM or PE in the hierarchical designs above.

6.5.2. Anycast SID for transport color endpoints (e.g., PEs)

The common Anycast SID technique may also be used for a redundant pair of PEs that share an identical set of service (VPN) attachments.

- For example, assume a node E2' paired with E2 above. Both PEs should be configured with the same static label/SID for the services (e.g., per-VRF VPN label/SID), and will advertise associated service routes with the Anycast IP as BGP next-hop.

- This design provides a convergence and recursive resolution benefit on an ingress PE or ABR similar to the egress ABR case above.

7. Routing Convergence

This section will analyze routing convergence.

8. VPN CAR

This section illustrates the extension of BGP CAR to address the VPN CAR requirement stated in Section 3.2 of [dskc-bess-bgp-car-problem-statement].

```
  CE1 -------------- PE1 -------------------- PE2 -------------- CE2 - V
```

- BGP CAR is enabled between CE1-PE1 and PE2-CE2

- BGP VPN CAR is enabled between PE1 and PE2

- Provider publishes intent 'low-delay' is mapped to color CP on its inbound peering links

- Within its infrastructure, Provider maps intent 'low-delay' to color CPT

- On CE1 and CE2, intent 'low-delay' is mapped to CC

(V, CC) is a Color-Aware route originated by CE2

```
   1.   CE2 sends to PE2     : [(V, CC), Label L1]  via CE2        with LCM (CP)
   2.   PE2 installs in VRF A: [(V, CC), L1]        via CE2        which resolves
 on (CE2, CP)
                                                                   / connected OI
F
   2.a. PE2 allocates VPN Label L2 and programs swap entry for (V, CC)
   3.   PE2 sends to PE1     : [(RD, V, CC), L2]    via PE2        with regular C
olor Extended
                                                                   Community (CPT
)
   4.   PE1 installs in VRF A: [(V, CC), L2]        via (PE2, CPT) steered on (PE
2, CPT)
   4.a. PE1 allocates Label L3 and programs swap entry for (V, CC)
   5.   PE1 sends to CE1     : [(V, CC), L3]        via PE1        without any LC
M
   6.   CE1 installs         : [(V, CC), L3]        via PE1        which resolves
 on (PE1, CC)
                                                                   / connected OI
F
   6.a. Label L3 is installed as the imposition label for (V, CC)
```

VPN CAR distribution for (RD, V, CC) requires a new SAFI that follows same VPN semantics as defined in [RFC4364], the difference being that the advertised routes carry CAR NLRI defined in Section 2.9.2 of this document.

VPN CAR NLRI with RD has the format shown 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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  NLRI Length  |  Key Length   |   NLRI Type   |Prefix Length  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |               Route Distinguisher                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |               Route Distinguisher                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |               IP Prefix (variable)                           //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |               Color (4 octets)                                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Followed by optional TLVs encoded as below:

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

where:

Route Distinguisher: 8 octet field encoded according to [RFC4364]

9. IANA Considerations

IANA is requested to assign SAFI value TBD1 (BGP CAR) and SAFI value TBD2 (BGP VPN CAR) from the "SAFI Values" sub-registry under the "Subsequent Address Family Identifiers (SAFI) Parameters" registry with this document as a reference.

9.1. BGP CAR NLRI Types Registry

IANA is requested to create a "BGP CAR NLRI Types" sub-registry under the "Border Gateway Protocol (BGP) Parameters" registry with this document as a reference. The registry is for assignment of the one octet sized code-points for BGP CAR NLRI types and populated with the values shown below:

Type	NLRI Type	Reference
0	Reserved (not to be used)	[This document]
1	Color-Aware Routes NLRI	[This document]
2-255	Unassigned	

Allocations within the registry are to be made under the "Specification Required" policy as specified in [RFC8126]).

9.2. BGP CAR NLRI TLV Registry

IANA is requested to create a "BGP CAR NLRI TLV Types" sub-registry under the "Border Gateway Protocol (BGP) Parameters" registry with this document as a reference. The registry is for assignment of the one octet sized code-points for BGP-CAR NLRI non-key TLV types and populated with the values shown below:

Type	NLRI Type	Reference
0	Reserved (not to be used)	[This document]
1	Label TLV	[This document]
2	Label Index TLV	[This document]
3	SRv6 SID TLV	[This document]
4-255	Unassigned	

Allocations within the registry are to be made under the "Specification Required" policy as specified in [RFC8126]).

9.3. 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.

9.4. BGP Extended Community Registry

IANA is requested to allocate the sub-type TBD2 for "Local Color Mapping (LCM)" under the "BGP Transitive Opaque Extended Community" registry under the "BGP Extended Community" parameter registry.

10. Acknowledgements

The authors would like to acknowledge the review and inputs from many people.TBD

11. References

11.1. Normative References

[I-D.ietf-bess-srv6-services]
Dawra, G., Filsfils, C., Talaulikar, K., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "SRv6 BGP based Overlay Services", draft-ietf-bess-srv6-services-07 (work in progress), April 2021.

[I-D.ietf-idr-bgp-ipv6-rt-constrain]
Patel, K., Raszuk, R., Djernaes, M., Dong, J., and M. Chen, "IPv6 Extensions for Route Target Distribution", draft-ietf-idr-bgp-ipv6-rt-constrain-12 (work in progress), April 2018.

[I-D.ietf-idr-tunnel-encaps]
Patel, K., Velde, G. V. D., Sangli, S. R., and J. Scudder, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-22 (work in progress), January 2021.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-15 (work in progress), April 2021.

[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-11 (work in progress), April 2021.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Garvia, P. C., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 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>.

[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>.

[RFC5512] Mohapatra, P. and E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", RFC 5512, DOI 10.17487/RFC5512, April 2009, <https://www.rfc-editor.org/info/rfc5512>.

[RFC5701] Rekhter, Y., "IPv6 Address Specific BGP Extended Community Attribute", RFC 5701, DOI 10.17487/RFC5701, November 2009, <https://www.rfc-editor.org/info/rfc5701>.

[RFC7311] Mohapatra, P., Fernando, R., Rosen, E., and J. Uttaro, "The Accumulated IGP Metric Attribute for BGP", RFC 7311, DOI 10.17487/RFC7311, August 2014, <https://www.rfc-editor.org/info/rfc7311>.

[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>.

[RFC8277] Rosen, E., "Using BGP to Bind MPLS Labels to Address Prefixes", RFC 8277, DOI 10.17487/RFC8277, October 2017, <https://www.rfc-editor.org/info/rfc8277>.

[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>.

[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>.

11.2. Informative References

[I-D.ietf-mpls-seamless-mpls] Leymann, N., Decraene, B., Filsfils, C., Konstantynowicz, M., and D. Steinberg, "Seamless MPLS Architecture", draft-ietf-mpls-seamless-mpls-07 (work in progress), June 2014.

[RFC3906] Shen, N. and H. Smit, "Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering Tunnels", RFC 3906, DOI 10.17487/RFC3906, October 2004, <https://www.rfc-editor.org/info/rfc3906>.

[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>.

[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>.

[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>.

[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. Illustrations of Service Steering

The following sub-sections illustrate example scenarios of Colored Service Route Steering over E2E BGP CAR resolving over different intra-domain mechanisms

The examples use MPLS/SR for the transport data plane. Scenarios specific to other encapsulations will be added in subsequent versions.

A.1. E2E BGP transport CAR intent realized using IGP FA

```
                              RD:V/v via E2
          +-----+             vpn label: 30030       +-----+
   ...... |S-RR1| <..................................|S-RR2| <.......
   :      +-----+             Color C1               +-----+        :
   :                                                                :
   :                                                                :
   :                                                                :
+-:-----------------------+----------------------+------------------:--+
| :                       |                      |                  :  |
| :                       |                      |                  :  |
| :   (E2,C1) via 121     |   (E2,C1) via 231    | (E2,C1)via E2    :  |
| :   L=168002,AIGP=110 +---+ L=168002,AIGP=10 +---+ L=0x3,LI=8002  :  |
| : |-------------------|121|<-----------------|231|<-------------| :  |
| : V LI=8002           +---+ LI=8002          +---+              | :  |
|----+                    |                      |               +-----|
| E1 |                    |                      |               | E2  |
|----+(E2,C1) via 122     |   (E2,C1) via 232    |  (E2,C1)via E2+-----|
|   ^ L=168002,AIGP=210 +---+ L=168002,AIGP=20 +---+ L=0x3        |    |
|   |----------------   |122|<-----------------|232|<-------------|    |
|     LI=8002           +---+ LI=8002          +---+ LI=8002           |
|                         |                      |                     |
|         ISIS SR         |      ISIS SR         |     ISIS SR         |
|         FA 128          |      FA 128          |     FA 128          |
+-------------------------+----------------------+---------------------+
iPE                     iABR                   eABR              ePE

+------+                  +------+
|168121|                  |168231|
+------+                  +------+
+------+                  +------+                 +------+
|168002|                  |168002|                 |168002|
+------+                  +------+                 +------+
+------+                  +------+                 +------+
|30030 |                  |30030 |                 |30030 |
+------+                  +------+                 +------+
```

Figure 6: BGP FA Aware transport CAR path

Use case: Provide end to end intent for service flows.

- o With reference to the topology above:

    - * IGP FA 128 is running in each domain.

    - * Egress PE E2 advertises a VPN route RD:V/v colored with (color extended community) C1 to steer traffic to BGP transport CAR (E2, C1). VPN route propagates via service RRs to ingress PE E1.

  - BGP CAR route (E2, C1) with next-hop, label-index and label as shown above are advertised through border routers in each domain. When a RR is used in the domain, ADD-PATH is enabled to advertise multiple available paths.

  - Local policy on each hop maps intent C1 to resolve CAR route next-hop over IGP FA 128 of the domain. AIGP attribute influences BGP CAR route best path decision as per [RFC7311]. BGP CAR label swap entry is installed that goes over FA 128 LSP to next-hop providing intent in each IGP domain. Update AIGP metric to reflect FA 128 metric to next-hop.

  - Ingress PE E1 learns CAR route (E2, C1). It steers colored VPN route RD:V/v into (E2, C1)

- Important:

  - IGP FA 128 top label provides intent in each domain.

  - BGP CAR label (e.g. 168002) carries end to end intent. Thus stitches intent over intra domain FA 128.

A.2. E2E BGP transport CAR intent realized using SR Policy

```
                              RD:1/8 via E2
          +-----+             vpn label: 30030       +-----+
   ...... |S-RR1| <..................................|S-RR2| <......
   :      +-----+             Color C1               +-----+        :
   :                                                                :
   :                                                                :
   :                                                                :
+-:-----------------------+----------------------+------------------:-+
| :                       |                      |                  : |
| :                       |                      |                  : |
| :  <-(E2,C1) via 121    |   <-(E2,C1) via 231  | <-(E2,C1)via E2  : |
| :                     +---+                  +---+                : |
| :  ------------------>|121|----------------->|231|--------------| : |
| : | SR policy(C,121)  +---+ SR policy(C1,231)+---+ SR policy    v : |
|----+                    |                      |   (C1,E2)      +---|
| E1 |                    |                      |                |E2 |
|----+ <-(E2,C1) via 122  |  (E2,C1) via 232     | <-(E2,C1)via E2+---|
|   |                   +---+                  +---+               ^  |
|    ------------------>|122|----------------->|232|---------------|  |
|    SR policy(C,122)   +---+ SR policy(C1,232)+---+ SR policy(C1,E2) |
|                         |                      |                    |
|                         |                      |                    |
|         ISIS SR         |      ISIS SR         |     ISIS SR        |
+-------------------------+----------------------+--------------------+
iPE                     iABR                     eABR                ePE
```

Figure 7: BGP SR policy Aware transport CAR path

Use case: Provide end to end intent for service flows

- o With reference to the topology above:
  - * SR Policy provide intra domain intent.
  - * Egress PE E2 advertises a VPN route RD:V/v colored with (color extended community) C1 to steer traffic to BGP transport CAR (E2, C1). VPN route propagates via service RRs to ingress PE E1.
  - * BGP CAR route (E2, C1) with next-hop, label-index and label as shown above are advertised through border routers in each domain. When a RR is used in the domain, ADD-PATH is enabled to advertise multiple available paths.
  - * Local policy on each hop maps intent C1 to resolve CAR route next-hop over an SR policy(C1, next-hop). BGP CAR label swap entry is installed that goes over SR policy segment list.

   - Ingress PE E1 learns CAR route (E2, C1). It steers colored VPN route RD:V/v into (E2, C1).

- Important:

   - SR policy provides intent in each domain.

   - BGP CAR label (e.g. 168002) carries end to end intent. Thus stitches intent over intra domain SR policies.

A.3. BGP transport CAR intent realized in a section of the network

```
                              RD:1/8 via E2
          +-----+             vpn label: 30030       +-----+
   ...... |S-RR1| <..................................|S-RR2| <.......
   :      +-----+             Color C1               +-----+        :
   :                                                                :
   :                                                                :
   :                                                                :
+-:-----------------------+----------------------+------------------:--+
| :                       |                      |                  :  |
| :                       |                      |                  :  |
| :   (E2,C1) via 121     |  (E2,C1) via 231     | (E2,C1) via E2   :  |
| :   L=168002,AIGP=1110+---+L=168002,AIGP=1010+---+ L=0x3          :  |
| : |-------------------|121|<-----------------|231|<-------------| :  |
| : V LI=8002           +---+ LI=8002          +---+              | :  |
|----+                    |                      |               +-----|
| E1 |                    |                      |               | E2  |
|----+(E2,C1) via 122     |  (E2,C1) via 232     | (E2,C1) via E2+-----|
|   ^ L=168002,AIGP=1210+---+L=168002,AIGP=1020+---+ L=0x3       |     |
|   |----------------   |122|<-----------------|232|<-------------|     |
|     LI=8002           +---+ LI=8002          +---+                   |
|                         |                      |                     |
|         ISIS SR         |       ISIS SR        |     ISIS SR         |
|         FA 0            |       FA 128         |     FA 0            |
|         Access          |       Core           |     Access
+-------------------------+----------------------+---------------------+
iPE                     iABR                       eABR              ePE

+------+                  +------+
|160121|                  |168231|
+------+                  +------+
+------+                  +------+                 +------+
|168002|                  |168002|                 |160002|
+------+                  +------+                 +------+
+------+                  +------+                 +------+
|30030 |                  |30030 |                 |30030 |
+------+                  +------+                 +------+
```

Figure 8: BGP Hybrid FA Aware transport CAR path

Use case: Provide intent for service flows only in Core domain.

o With reference to the topology above:

   * IGP FA 128 is only enabled in Core (e.g. WAN network). Access only has base algo 0.

   * Egress PE E2 advertises a VPN route RD:V/v colored with (color extended community) C1 to steer traffic to BGP transport CAR

(E2, C1). VPN route propagates via service RRs to ingress PE E1.

  - BGP CAR route (E2, C1) with next-hop, label-index and label as shown above are advertised through border routers in each domain. When a RR is used in the domain, ADD-PATH is enabled to advertise multiple available paths.

  - Local policy on 231 and 232 maps intent C1 to resolve CAR route next-hop over IGP base algo 0 in right access domain. BGP CAR label swap entry is installed that goes over algo 0 LSP to next-hop. Update AIGP metric to reflect algo 0 metric to next-hop with an additional penalty.

  - Local policy on 121 and 122 maps intent C1 to resolve CAR route next-hop learnt from Core domain over IGP FA 128. BGP CAR label swap entry is installed that goes over FA 128 LSP to next-hop providing intent in Core IGP domain.

  - Ingress PE E1 learns CAR route (E2, C1). It maps intent C1 to resolve CAR route next-hop over IGP base algo 0. It steers colored VPN route RD:V/v into (E2, C1)

- Important:

  - IGP FA 128 top label provides intent in Core domain.

  - BGP CAR label (e.g. 168002) carries intent from PEs which is realized in core domain

A.4. Transit network domains that do not support CAR

- In a brownfield deployment, color-aware paths between two PEs may need to go through a transit domain that does not support CAR. Example include an MPLS LDP network with IGP best-effort; or a BGP-LU based multi-domain network. MPLS LDP network with best effort IGP can adopt above scheme. Below is the example for BGP LU.

- Reference topology:

```
E1 --- BR1 --- BR2 ......... BR3 ---- BR4 --- E2
    Ci           <----LU---->              Ci
```

  - Network between BR2 and BR3 comprises of multiple BGP-LU hops (over IGP-LDP domains).

  - E1, BR1, BR4 and E2 are enabled for BGP CAR, with Ci colors

   - BR1 and BR2 are directly connected; BR3 and BR4 are directly connected

- BR1 and BR4 form an over-the-top peering (via RRs as needed) to exchange BGP CAR routes

- BR1 and BR4 also form direct BGP-LU sessions to BR2 and BR3 respectively, to establish labeled paths between each other through the BGP-LU network

- BR1 recursively resolves the BGP CAR next-hop for CAR routes learnt from BR4 via the BGP-LU path to BR4

- BR1 signals the transport discontinuity to E1 via the AIGP TLV, so that E1 can prefer other paths if available

- BR4 does the same in the reverse direction

- Thus, the color-awareness of the routes and hence the paths in the data plane are maintained between E1 and E2, even if the intent is not available within the BGP-LU island

- A similar design can be used for going over network islands of other types

Appendix B. Color Mapping Illustrations

There are a variety of deployment scenarios that arise w.r.t different color mappings in an inter-domain environment. This section attempts to enumerate them and provide clarity into the usage of the color related protocol constructs.

B.1. Single color domain containing network domains with N:N color distribution

- All network domains (ingress, egress and all transit domains) are enabled for the same N colors.

   - A color may of course be realized by different technologies in different domains as described above.

- The N intents are both signaled end-to-end via BGP CAR routes; as well as realized in the data plane.

- Appendix A.1 is an example of this case.

B.2. Single color domain containing network domains with N:M color distribution

- Certain network domains may not be enabled for some of the colors, but may still be required to provide transit.

- When a (E, C) route traverses a domain where color C is not available, the operator may decide to use a different intent of color c that is available in that domain to resolve the next-hop and establish a path through the domain.

  * The next-hop resolution may occur via paths of any intra-domain protocol or even via paths provided by BGP CAR.

  * The next-hop resolution color c may be defined as a local policy at ingress or transit nodes of the domain.

  * It may also be automatically signaled from egress border nodes by attaching a color extended community with value c to the BGP CAR routes.

- Hence, routes of N colors may be resolved via a smaller set of M colored paths in a transit domain, while preserving the original color-awareness end-to-end.

- Any ingress PE that installs a service (VPN) route with a color C, must have C enabled locally to install IP routes to (E, C) and resolve the service route next-hop.

- A degenerate variation of this scenario is where a transit domain does not support any color. Appendix A.3 describes an example of this case.

B.3. Multiple color domains

When the routes are distributed between domains with different color-to-intent mapping schemes, both N:N and N:M cases are possible, although an N:M mapping is more likely to occur.

Reference topology:

```
D1 ----- D2 ----- D3
C1       C2       C3
```

- C1 in D1 maps to C2 in D2 and to C3 in D3

- BGP CAR is enabled in all three domains

The reference topology above is used to elaborate on the design described in Section 2.8

When the route originates in color domain D1 and gets advertised to a different color domain D2, following procedures apply:

- The original intent in the BGP CAR route is preserved; i.e. route is (E, C1)

- A BR of D1 attaches LCM-EC with value C1 when advertising to a BR in D2

- A BR in D2 receiving (E, C1) maps C1 in received LCM-EC to local color, say C2

- Within D2, this LCM-EC value of C2 is used instead of the Color in CAR route NLRI (E, C1). This applies to all procedures described in the earlier section for a single color domain, such as next-hop resolution and installation of route and forwarding entries.

- A colored service route V/v originated in domain D1 with next-hop E and color C1 will also have its color extended-community value re-mapped to C2, typically at a service RR

- On an ingress PE in D2, V/v will resolve via C2

- When a BR in D2 advertises the route to a BR in D3, the same process repeats.

Authors' Addresses

Dhananjaya Rao
Cisco Systems
USA

Email: dhrao@cisco.com

Swadesh Agrawal
Cisco Systems
USA

Email: swaagraw@cisco.com

Clarence Filsfils
Cisco Systems
Belgium

Email: cfilsfil@cisco.com

Ketan Talaulikar
Cisco Systems
India

Email: ketant@cisco.com

Dirk Steinberg
Lapishills Consulting Limited
Germany

Email: dirk@lapishills.com

Luay Jalil
Verizon
USA

Email: luay.jalil@verizon.com

Yuanchao Su
Alibaba, Inc

Email: yitai.syc@alibaba-inc.com

Jim Guichard
Futurewei
USA

Email: james.n.guichard@futurewei.com

Keyur Patel
Arrcus, Inc
USA

Email: keyur@arrcus.com

Haibo Wang
Huawei Technologies
China

Email: rainsword.wang@huawei.com

BESS WorkGroup
Internet-Draft
Intended status: Informational
Expires: August 26, 2021

D. Rao
S. Agrawal
C. Filsfils
K. Talaulikar
Cisco Systems
B. Decraene
Orange
D. Steinberg
Steinberg Consulting
L. Jalil
Verizon
J. Guichard
Futurewei
W. Henderickx
Nokia

February 22, 2021

BGP Color-Aware Routing Problem Statement
draft-dskc-bess-bgp-car-problem-statement-02

Abstract

This document explores the scope, use-cases and requirements for a BGP based routing solution to establish end-to-end intent-aware paths across a multi-domain service provider network environment.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

1.1. Objective

This document explores the scope, use-cases and requirements for a BGP based routing solution to establish end-to-end intent-aware paths across a multi-domain service provider network environment.

The targeted design outcome is to define the technology and protocol extensions that may be required in a manner that addresses the widest application.

To introduce the problem space that the document focuses on, let us start with the BGP-based delivery of an intent across several SR-MPLS/MPLS domains.

```
          +-----+                +-----+     V/v via PE2  +-----+
   .......|S-RR1|<...............|S-RR2|<.................|S-RR3| <..
   :      +-----+                +-----+     Color C      +-----+   :
   :                                                                :
   :                                                                :
+--:----------+-------------+--------------+-------------+----------:--+
|  :          |             |              |             |          :  |
|  :          |             |              |             |          :  |
|  :        +---+         +---+          +---+         +---+        :  |
|  :        |121|         |231|          |341|         |451|        :  |
|  :        +---+         +---+          +---+         +---+        :  |
|----+        |             |              |             |        +----|
| PE1|        |             |              |             |        |PE2 |
|----+        |             |              |             |        +----|
|           +---+         +---+          +---+         +---+           |
|           |122|         |232|          |342|         |452|           |
|           +---+         +---+          +---+         +---+           |
|  Access     |   Metro     |   Core       |   Metro     |   Access    |
|  domain 1   |   domain 2  |   domain 3   |   domain 4  |   domain 5  |
+-------------+-------------+--------------+-------------+-------------+
iPE         iBRM          iBRC           eBRC           eBRM         ePE
```

Figure 1: Reference large-scale multi-domain network topology

The figure above shows a reference large-scale multi-domain network topology. PE1 and PE2 are PEs; the other nodes are border routers

(BR) between domains in different tiers of the network. A VPN route is advertised via service RRs (S-RR) between an egress PE (PE2) and an ingress PE (PE1).

BGP must provide reachability from PE1 to PE2 based on various intent. For instance, BGP may provide reachability to PE2 using either low latency or best effort.

A VPN route having a requirement of low latency routing will select the BGP reachability information to PE2 that is based on low latency.

The problem space is then widened to include any intent (including NFV chains and their location), any dataplane and the application of the intent-based routing to the Service/VPN routes. All of this is detailed in the rest of the document.

1.2. State-of-the-art

The following solution is widely deployed - [I-D.ietf-spring-segment-routing-policy]:

- In reference figure above, an Egress PE PE2 advertises a BGP VPN route V/v with a BGP Color Extended Community C [I-D.ietf-idr-tunnel-encaps] to indicate the service intent that PE2 requests for the traffic bound to V/v. Note: The Color Extended Community may be applied to any BGP service route. For simplicity in this document, we will use a VPN route example.

- An ingress PE1 steers V-destined packets onto an SR Policy bound to (C, PE2).

- C may express any of the following requirements:

  - Minimization of a cost metric vs a latency metric.

  - Exclusion/Inclusion of SRLG and/or Link Affinity.

    - In inter-domain context, exclusion/inclusion of entire domains.

  - Inclusion of virtual network function chains [I-D.ietf-spring-sr-service-programming].

- An SR-PCE (or a set of them) computes the end-to-end path and installs it at PE1 as an SR Policy. The end-to-end path may seamlessly cross multiple domains.

The SR-PCE solution being defined at the IETF [RFC8664]and being widely deployed is reminded in this introduction as a useful "state-of-the-art" context to consider when defining the BGP-based alternative solution.

1.2.1. Color

The solution must reuse the Color concept defined in [I-D.ietf-spring-segment-routing-policy]. The color is a 32-bit numerical value that, today, associates an SR-policy with an intent (e.g., low latency).

1.2.2. Colored vs Color-Aware

The solution must support the ability to distinguish BGP routes that require the usage of a particular intent from BGP routes that are actually satisfying a particular intent. Hence, this document defines the notion of colored and color-aware routes.

- Colored: Egress PE PE2 colored its BGP VPN route V/v to indicate the intent that it requests for the traffic bound to V/v.

- Color-Aware: A new BGP solution which signals multiple "ways" to reach a given destination (e.g. PE2)

- Steering a colored VPN route to a color-aware route

  - If PE2 signals a VPN route V/v with color C

  - If PE1 installs that VPN route

  - If PE1 learns about a BGP Color-Aware Route R/r to PE2 for color C

  - Then PE1 steers packets destined to V/v via R/r

- Note the similarity with the state-of-the-art reference:

  - The steering onto an SR Policy bound to (C, PE2) is replaced by the steering on a Color-Aware BGP route (C, PE2)

  - The data model is the same "resolution via (C, PE2)"

  - The difference is how the (C, PE2) path is obtained: BGP signaling vs SR-PCE signaling

1.2.3. Per-Destination and Per-Flow Steering

Ingress PE PE1 steers packets destined for a service (VPN) route V/v via BGP Color-Aware Route R/r to PE2

- o Per-Destination Steering: Incoming packets on PE1 match BGP service route V/v to be steered based on the destination IP address of the packets.

- o Per-Flow Steering: Incoming packets on PE1 match BGP service route V/v to be steered based on the combination of the destination IP address and additional elements in the packet header (i.e., IP flow). Such a packet lookup may recurse on a forwarding array where some of the entries are BGP color-aware routes to PE2. A given flow is mapped to a specific entry in this array i.e. via a specific BGP color-aware route to PE2.

1.3. Why a BGP-based alternative is needed

- o An operator with an existing Seamless-MPLS/BGP-LU deployment [I-D.ietf-mpls-seamless-mpls] may consider a BGP based extension as a more incremental approach.

- o There may be an expectation that BGP would support a larger scale.

- o Opacity of a remote domain due to trust boundaries within an inter-domain construction.

1.4. Color Domains

With the use of Color to represent intent, it is useful to describe the concept of a color domain distinct from a network domain.

- o Domain: A domain (or network domain) refers to a unit of isolation or hierarchy in the network topology; for example, access, metro and or core domains. From a routing perspective, a domain may have a distinct IGP area or instance; or a distinct BGP ASN.

- o Color Domain: A color domain may represent a collection of one or more network domains with a single, consistent color/intent mapping.

- o Color re-mapping may happen at color domain boundaries.

- o Deployments under a single authority are expected to use the same color/intent mapping across all network domains.

A solution must distinguish the actual protocol boundaries (IGP, ASN) from the color domain boundaries.

1.5. BGP Color-Aware Routing

A BGP solution that is solving this problem statement is called BGP Color-Aware Routing, and is referred to as BGP CAR in this document.

2. Intent bound to a Color

The BGP CAR solution must support the following intents bound to a color:

- Minimization of a cost metric vs a latency metric
  - Minimization of different metric types, static and dynamic
- Exclusion/Inclusion of SRLG and/or Link Affinity and/or minimum MTU/number of hops
- Bandwidth management
- In the inter-domain context, exclusion/inclusion of entire domains, and border routers
- Inclusion of one or several virtual network function chains
  - Located in a regional domain and/or core domain, in a DC
- Localization of the virtual network function chains
  - Some functions may be desired in the regional DC or vice versa
- Per-Destination and Per-Flow steering

3. BGP CAR Use-cases

The BGP CAR route may be a transport route or a service route (in this document, we use the term VPN instead of service for simplicity).

3.1. BGP Transport CAR

- Transport Intent
  - Intent-aware routing between PEs connected across multiple transit domains

- + Set up BGP based end-to-end paths stitching intent-aware intra-domain segments

- o The network diagram below illustrates the reference network topology used in this section for Transport CAR:

```
        +-----+                +-----+                  +-----+
   .....|S-RR1| <............. |S-RR2| <............... |S-RR3| <...
   :    +-----+                +-----+                  +-----+    :
   :                                                               :
   :                                                               :
   :                                                               :
+--:--------------+       +-----------------+       +--------------:--+
|  :              |       |                 |       |              :  |
|  :              |       |                 |       |              :  |
|  :          +---|  D=20 |---+         +---|  D=25 |---+          :  |
|  :          |121|-------|211|         |231|-------|321|          :  |
|  :          +---| \   / |---+         +---| \   / |---+          :  |
|----+             |  \ /  |                 |  \ /  |            +----|
|PE11|             |   V   |                 |   V   |            |PE31|
|----+             |  / \  |                 |  / \  |            +----|
|             +---| /   \ |---+         +---| /   \ |---+             |
|             |122|-------|212|         |232|-------|322|             |
|             +---|  D=15 |---+         +---|  D=10 |---+             |
|                 |       |                 |       |                 |
|    Domain 1     |       |    Domain 2     |       |    Domain 3     |
+-----------------+       +-----------------+       +------------------+
```

Figure 2: Transport CAR Reference Topology

The following network design assumptions apply to the reference topology above, as an example:

* Independent ISIS/OSPF SR instance in each domain.

* eBGP peering link between ASBRs (121-211, 121-212, 122-211, 122-212, 231-321, 231-322, 232-321 and 232-322).

* Peering links have equal cost metric.

* Peering links have delay configured or measured as shown by "D". D=50 for cross peering links.

* VPN service is running from PE31 to PE11 via service RRs (S-RRn in figure).

- o The following sections illustrate a few examples of intent use-cases applicable to transport routes.

3.1.1. Use-case of minimization of a cost metric vs a latency metric

- In the reference topology of Figure 2

  Each domain has Algo 0 and Flex Algo 128

  Algo 0 is for minimum cost metric(cost optimized).

  Flex Algo 128 definition is for minimum delay (low latency).

- Cost Optimized

  - Color C1 - Minimum cost intent. (Here, a BGP CAR route with Color C1 is being used, instead of BGP-LU.)

  - On PE11, VPN routes colored with C1 are steered via (C1, PE31) BGP CAR route

    - BGP CAR for C1 sets up path(s) between PEs for end-to-end minimum cost.

    - (2) These paths traverse over intra-domain Algo 0 in each domain and account for the peering link cost between ASBRs.

    - Example: PE11 learns (C1, PE31) CAR route via several equal paths:

      1. One such path is through FA0 to node 121, links 121-211, FA0 to 231, link 231-321, FA0 to PE31

      2. Another such path is through FA0 to node 122, link 122-212, FA0 to 232, link 232-322, FA0 to PE31.

- Minimize latency

  - Color C2 - Minimum latency intent.

  - On PE11, VPN routes colored with C2 are steered via (C2, PE31) BGP CAR route.

    - BGP CAR for C2 advertises paths between PEs for minimum end-to-end delay.

    - (2) These paths traverse over intra-domain Flex Algo 128 in each domain and account for peering link delay between ASBRs.

- (3) Example: PE11 learns (C2, PE31) BGP CAR route and best path is through FA128 to node 122, link 122-212, FA128 to 232, link 232-322, FA128 to PE31.

3.1.2. Use-case of exclusion/inclusion of link affinity

- Color C3 - Intent to Minimize cost metric and avoid purple links
- In the reference topology of Figure 2

  Each domain has Flex Algo 129 and some links have purple affinity.

  Flex Algo 129 definition is set to minimum cost metric and avoid purple links (within domain).

  Peering cross links are colored purple by policy.

- On PE11, VPN routes colored with C3 are steered via (C3, PE31) BGP CAR route.

  * BGP CAR for C3 sets up paths between PEs for minimum end-to-end cost and avoiding purple link affinity.

  * These paths traverse over intra domain Flex Algo 129 in each domain and accounts for peering link cost between ASBR and avoiding purple links.

  * Example: PE11 learns (C3, PE31) BGP CAR route via 2 paths.

    1. First path is through FA 129 to node 121, link 121-211, FA129 to 231, link 231-321, FA129 to PE31.

    2. Second path is through FA129 to node 122, link 122-212, FA129 to 232, link 232-322, FA129 to PE31.

3.1.3. Use-case of exclusion/inclusion of domains

```
         +-----+                +-----+                 +-----+
     ....|S-RR1| <............. |S-RR2| <.............. |S-RR3| <....
     :   +-----+                +-----+                 +-----+     :
     :                                                              :
     :                      +----------------+                      :
     :                      |                |                      :
  +--:--------------+       |---+        +---|       +--------------:--+
  |  :              |   |---|211|        |241|---|   |              :  |
  |  :              |   |   |---+        +---|   |   |              :  |
  |  :          +---|   |   |    Domain 2    |   |   |---+          :  |
  |  :          |121|---|   +----------------+   |---|421|          :  |
  |  :          +---|                                |---+          :  |
  |----+            |                                |            +----|
  |PE11|            |                                |            |PE41|
  |----+            |                                |            +----|
  |             +---|                                |---+             |
  |             |131|---|   +----------------+   |---|431|             |
  |             +---|   |   |                |   |   |---+             |
  |                 |   |   |---+        +---|   |   |                 |
  |    Domain 1     |   |---|311|        |341|---|   |      Domain 4   |
  +-----------------+       |---+        +---|       +-----------------+
                            |   Domain 3     |
                            +----------------+
```

Figure 3

Color C4 - Avoid sending selected traffic via Domain 3

- VPN routes advertised from PEs with Color C4
- BGP CAR for Color C4 should only set up paths between PE11 and PE41 that exclude Domain 3

3.1.4. Use-case of virtual network function chains in local and core domains

```
          ____                      ____
         /    \                    /    \
        | NFV1 |                  | NFV2 |
         \    /                    \    /
  +---------------------+  +--------------------+  +-------------------+
  |      |E11|          |  |       |E21|        |  |                   |
  |      +---+          |  |       +---+        |  |                   |
  |                     |  |                    |  |                   |
  |                     |  |                    |  |                   |
  |----+              +------+                +------+            +----|
  |PE11|              | 121  |                | 231  |            |PE31|
  |----+              +------+                +------+            +----|
  |                     |  |                    |  |                   |
  |                     |  |                    |  |                   |
  |                     |  |                    |  |                   |
  |                     |  |                    |  |                   |
  |      Domain 1       |  |      Domain 2      |  |    Domain 3       |
  +---------------------+  +--------------------+  +-------------------+
```

Figure 4

o Color intent

   * C5 - Routing via min-cost paths

   * C6 - Routing via a local NFV service chain situated at E11

   * C7 - Routing via a centrally located NFV service chain situated at E21

o Forwarding of packets from PE11 towards PE31:

   * (C5, PE31) mapped packets are sent via nodes 121, 231 to PE31

   * (C6, PE31) mapped packets are sent to E11 and then post-service chain, via 121, 231 to PE31

   * (C7, PE31) mapped packets are sent via 121 to E21 and then post-service chain, via 231 to PE31

3.2. BGP VPN CAR

o VPN (Service layer) intent

   * Extend the signaling of intent awareness end-to-end: CE site to CE site across provider networks

    - Provide ability for a CE to select paths through specific PEs for a given intent

      - Example-1: Certain intent in transport not available via specific PEs

      - Example-2: Certain CE-PE connection does not support specific intent

      - Example-3: Site access via certain CE does not support specific intent. For instance, link connecting a specific CE to a DC hosting loss-sensitive service may have better quality than a link from another CE

    - Provide ability for a CE to send traffic indicating a specific intent (via suitable encapsulation) to the PE for optimal steering.

  - Intent aware routing support for multiple service (VPN) interworking models

    - Beyond options such as iBGP or Inter-AS Option C that inherently extend from PE to PE

      1. Inter-AS Option A

      2. Inter-AS Option B

      3. GW based interworking(L3VPN, EVPN)

    - Interworking with existing L3VPN deployments, both PEs and CEs

- The network diagram below illustrates the reference network topology used in this section for VPN CAR.

```
        +-------------------+           +-------------------+
        |   ....|S-RR|....  |           |  ....|S-RR|.....  |
        |   :   +----+   :  |           |  :   +----+    :  |
        |   :    :  :    :  |           |  :    :  :     :  |
        |----+   :  :   +---|   D=20    |---+   :  :   +----|
       /|PE11|   :  :   |121|-----------|211|   :  :   |PE21|\
 D=25/  |----+   :  :   +---| X       X |---+   :  :   +----|  \ D=25
   /    |        :  :       |   X   X   |       :  :        |    \   V/24
CE1     |        :  :       |     X D=50|       :  :        |     CE2<---
   X    |        :  :       |   X   X   |       :  :        |    X
D=15 X  |----+   :  :   +---| X       X |---+   :  :   +----|  X D=10
       X|PE12|...:  :...|212|-----------|232|...:  :...|PE22|X
        |----+          +---|   D=10    |---+          +----|
        |                   |           |                   |
        |     AS 1          |           |     AS 2          |
        +-------------------+           +-------------------+
```

Figure 5: VPN CAR reference topology

The following network design assumptions apply to the reference topology above, as an example:

* Independent ISIS/OSPF SR instance in each AS.

* eBGP peering link between VPN ASBRs 121-211, 121-212, 122-211, 122-212.

* VPN service is running between PEs via service RRs in each AS to local ASBRs. Between ASBRs, its Option-B i.e. next hop self for VPN SAFI.

* CE1 is dual homed to PE11 and PE12. Similarly, CE2 is dual homed to PE21 and PE22.

* Peering links have equal cost metric

* Peering links have delay configured or measured as shown by "D".

* CE2 advertises prefix V/24 to CE1. It is advertised as RD:V/24 between PEs, including color-awareness

o The following sections illustrate a few examples of intent use-cases applicable to VPN (service) routes.

3.2.1. Use-case of minimization of a cost metric vs a latency metric

- In the reference topology of Figure 5

  Each AS has Flex Algo 0 and 128.

  Flex Algo 0 is for minimum cost metric(cost optimized).

  Flex Algo 128 definition is for minimum delay (low latency).

- Cost Optimized

  * Color C1 - Minimum cost intent.

  * On CE1, flows requiring cost optimized paths to V/24 are steered over (C1, V/24) route.

    + BGP CAR for C1 sets up paths between CEs for minimum end-to-end cost.

    + This advertisement needs BGP CAR between PE-CE for V/24 prefix and color C1 awareness.

    + It also needs BGP VPN CAR between PEs and ASBRs for RD:V/24 prefix and color C1 awareness (C1, RD:V/24).

    + Paths traverse over PE-CE links, intra-domain Flex Algo 0 in each AS and peering links between ASBRs, minimizing cost for VPN.

    + Example: CE1 learns (C1, V/24) CAR route through several equal cost paths:

      1. One path is through link CE1-PE11, FA0 to 121, link 121-211, FA0 to PE21 and link PE21-CE2.

      2. Another such path is through CE1-PE12, FA0 to node 122, link 122-212, FA0 to PE22, link PE22-CE2.

- Minimize latency

  * Color C2 - Minimum latency intent

  * On CE1, flows requiring low latency paths to prefix V/24 are steered over (C2, V/24) CAR route.

    + BGP CAR for C2 sets up paths between CEs for minimum end-to-end delay.

    - This advertisement needs BGP CAR between PE-CE for V/24 prefix and color C2 awareness.

    - It also needs BGP VPN CAR between PEs and ASBR for RD:V/24 prefix and color C2 awareness (C2, RD:V/24).

    - Paths traverse over intra-domain Flex Algo 128 in each AS and accounts for inter ASBR link delays and PE-CE link delays for the VPN.

    - Example: CE1 learns (C2, V/24) CAR best route through link CE1-PE12, FA128 to 122, link 122-212, FA128 to PE22 and link PE22-CE2.

3.2.2. Use-case of exclusion/inclusion of link affinity

- Color C3 - Intent to Minimize cost metric and avoid purple links

- In the reference topology of Figure 5

  Each AS has Flex Algo 129 and some links have purple affinity.

  Flex Algo 129 definition is set to minimum cost metric and avoid purple links (within AS).

  ASBR cross links are colored purple by policy. Bottom PE-CE links are colored purple as well by policy

- On CE1, flows requiring minimum cost path avoiding purple links to V/24 are steered over (C3, V/24) BGP CAR route.

  * BGP CAR for C3 setup paths between CEs for minimum end-to-end cost and avoiding purple link affinity.

  * This advertisement needs BGP CAR between PE-CE for V/24 prefix and color C3 awareness

  * It also needs BGP VPN CAR between PEs and ASBRs for RD:V/24 prefix and color C3 awareness (C3, RD:V/24).

  * The path avoids purple PE-CE links, traverses over intra-domain Flex Algo 129 in each AS and avoids purple links between VPN ASBRs.

  * Example: CE1 learns (C3, V/24) CAR route through link CE1-PE11, FA129 to 121, link 121-211, FA129 to PE21 and link PE21-CE2.

3.2.3. Use-case of virtual network function chains in local and core domains

```
                               +-----+
       ........................|S-RR | <.................
       :                       +-----+ <...........     :
       :                                           :    :
       :                                           :    :
       :  ___     ___           ___                :    :
       : /   \   /   \         /   \               :    :
       :| S1  | | S2  |       | S3  |              :    :
       : \   /   \   /         \   /               :    :
     +-:---------------+  +--------------+  +------:----:--+
     | :  |E11|  |E12| |  |    |E21|     |  |      :    :  |
     | :  +---+  +---+ |  |    +---+     |  |      :  +----|<-(V1/24,C1)
     | :               |  |              |  |      :  |PE31|--CE2
     | V               |  |              |  |      :  +----|
     |----+          +------+            +------+  :       |
CE1--|PE11|          | 121  |            | 231  |  :       |
     |----+          +------+            +------+  :  +----|<-(V2/24/C2)
     |                 |  |              |  |      :..|PE32|--CE3
     |                 |  |              |  |         +----|
     |                 |  |              |  |              |
     |    Domain 1     |  |   Domain 2   |  |   Domain 3   |
     +-----------------+  +--------------+  +--------------+
```

Figure 6

- Color intent

  * C1 - Routing via NFV service chain comprising of [S1, S2] attached to E11 and E12

  * C2 - Routing via NFV service [S3] attached to E21

- CE1, CE2, CE3 are sites of VPN1.

- Prefix V1/24 colored with C1 from CE2, and advertised as RD:V1/24 with C1 by PE31 to PE11 via S-RR

- Prefix V2/24 colored with C2 from CE3, and advertised as RD:V2/24 with C2 by PE32 to PE11 via SS-RR

- From PE11:

  - [V1/24, C1] mapped packets are sent via S1, S2 and then routed to PE31, CE2

  - [V2/24, C2] mapped packets are sent via S3 and then routed to PE32, CE3

4. Deployment Requirements

- Co-existence, compatibility and interworking with currently deployed SR-PCE based multi-domain color-aware solution

- Support different multi-domain deployment designs

  - Multiple IGP domains within a single AS (Seamless MPLS)

    - Inter-connect at node level (ABR)

  - Multiple BGP AS domains

    - Inter-connect via peering links (ASBR)

- Support end-to-end path crossing transport domains with different technologies and encapsulations

  - LDP-MPLS

  - RSVP-TE-MPLS

  - SR-MPLS

  - SRv6

  - IPv4/IPv6

- Support interworking between domains with different encapsulations (e.g, SR-MPLS and SRv6)

- Support multiple transport encapsulations within a domain for co-existence and migration

- Provide a BGP-based control-plane solution for the use-case illustrated in [RFC8604] together with deployment design guidelines for the leverage of anycast and binding SIDs.

5. Scalability

5.1. Scale Requirements

- o Support for massive scaled transport network
    - * Number of Remote PE’s: >= 300k
    - * Number of Colors C: >= 5
- o Scalable MPLS dataplane solution
    - * With one label per (C, Remote PE), the 1M MPLS dataplane does not work.
    - * A notion of hierarchy or segment list is required.
        - + E.g. the SR-PCE builds the end-to-end path as a list of segments such that no single node needs to support a data-plane scaling in the order of (Remote PE * C)
        - + The solution is thus not a direct extension of BGP-LU
    - * Additionally, PE and transit nodes (ABRs) may be devices with limited forwarding table space
    - * Devices may have constraints on packet processing (e.g., label operations, number of labels pushed) and performance
- o Ability to abstract the topology from remote domains - for scale, stability and faster convergence
    - * Abstracting PE and/or ABR related state and network events
- o Support for an Emulated-PULL model for the BGP signaling
    - * The SR-PCE solution natively supports a PULL model: when PE1 installs a VPN route V/v via (C, PE2), PE1 requests its serving SR-PCE to compute the SR Policy to (C, PE2). I.e. PE1 does not learn unneeded SR policies.
    - * BGP Signaling is natively a PUSH model.
    - * Emulated-PULL refers to the ability for a BGP CAR node PE1 to "subscribe" to (C, PE2) route such that only the related paths are signaled to PE1.

- The subscription and related filtering solution must apply to any BGP CAR node

  - Transport CAR routes

    1. Ability for a node (PE/ABR/RR) to signal interest for routes of specific colors.

    2. PEs only learn routes that they need - remote VPN endpoints (PEs/ASBRs) or transit nodes (ABRs, ASBRs).

    3. ABRs also only learn and propagate routes they need locally in domain

  - Service/VPN CAR routes

    1. Ability for a node (PE) to signal interest for a specific (Egress PE, Color) transport route

    2. CEs learn routes that they need - interested colors

    3. PEs learn routes that they need - interested VPNs, colors

  - Automation of the subscription/filter route

    1. Similar to the SR-PCE solution, when an ingress PE1 installs VPN V/v via (C, PE2), PE1 originates its subscription/filter route for (C, PE2).

  - Efficient propagation and processing of subscription/filter routes.

  - Ability to perform aggregation and suppression of subscription/filter routes at nodes in the route propagation path to reduce explosion and churn in propagation of the filter routes themselves.

  - The solution may be optional for networks that do not have the large scaling requirements

5.2. Scale Analysis

It is useful to analyze the multiple scaling requirements and specifically the data plane constraints in the context of a few common reference designs and use-cases.

A couple of example scenarios are listed below for reference.

- Seamless-MPLS design, with IGP Flex-Algo in each domain

```
                                    +-----+
     .............................. |S-RR | <........................
     :                              +-----+                         :
     :                                                              :
     :                                                              :
  +--:-----------------+  +--------------------+  +-----------------:--+
  |  :                 |  |                    |  |                 :  |
  |  :                 |  |                    |  |                 :  |
  |  :               +------+                +------+               :  |
  |  :               | 121  |                | 231  |               :  |
  |  V               +------+                +------+               :  |
  |----+               |  |                    |  |               +----|
  |PE11|               |  |                    |  |               |PE31|
  |----+               |  |                    |  |               +----|
  |                  +------+                +------+                  |
  |                  | 122  |                | 232  |                  |
  |                  +------+                +------+                  |
  |                    |  |                    |  |                    |
  |      Domain 1      |  |      Domain 2      |  |     Domain 3       |
  +--------------------+  +--------------------+  +--------------------+
```

Figure 7

- Inter-AS Option C VPN design, with IGP Flex-Algo in each domain, and eBGP peering between domains

```
        +-----+                +-----+                  +-----+
    ....|S-RR1| <............. |S-RR2| <............... |S-RR3| <...
    :   +-----+                +-----+                  +-----+     :
    :                                                               :
    :                                                               :
    :                                                               :
 +--:---------------+      +------------------+      +--------------:--+
 |  :               |      |                  |      |              :  |
 |  :               |      |                  |      |              :  |
 |  :           +---|      |---+          +---|      |---+          :  |
 |  :           |121|------|211|          |231|------|321|          :  |
 |  :           +---|      |---+          +---|      |---+          :  |
 |----+             |      |                  |      |            +----|
 |PE11|             |      |                  |      |            |PE31|
 |----+             |      |                  |      |            +----|
 |              +---|      |---+          +---|      |---+             |
 |              |122|------|212|          |232|------|322|             |
 |              +---|      |---+          +---|      |---+             |
 |                  |      |                  |      |                 |
 |    Domain 1      |      |    Domain 2      |      |    Domain 3     |
 +------------------+      +------------------+      +-----------------+
```

Figure 8

6. Network Availability

- o The BGP CAR solution should provide high network availability for typical deployment topologies, with minimum loss of connectivity in different network failure scenarios.

- o The network failure scenarios, applicable technologies and design options described in [I-D.ietf-mpls-seamless-mpls] should be used as a reference.

- o In the Seamless-MPLS reference topology in previous section:

  * Failure of intra-domain links should limit loss of connectivity (LoC) to < 50ms. E.g., PE11 to a P node (not shown), 121 to a P node in Domain1 or Domain2)

  * Failure of an intra-domain node (P node in any domain) should limit LoC to < 50ms

  * Failure of an ABR node (e.g., 121, 231) should limit LoC to < 1sec

  - Failure of a remote PE node (e.g., PE3) should limit LoC to < 1sec

- In the Inter-AS Option C VPN reference topology in previous section:

  - Failure of intra-domain links should limit LoC to < 50ms. E.g., PE11 to a P node (not shown), 121 to a P node in Domain1 or Domain2)

  - Failure of an intra-domain node (P node in any domain) should limit LoC to < 50ms

  - Failure of an ASBR node (e.g., 121, 211) should limit LoC to < 1sec

  - Failure of a remote PE node (e.g., PE3) should limit LoC to < 1sec

  - Failure of an external link (e.g., 121-211) should limit LoC to < 1sec

- The solution should explore and describe additional techniques and design options that are applicable to further improve handling of the failure cases listed above.

7. BGP Protocol Requirements

- Support signaling and distribution of different Color-Aware routes to reach a participating node, e.g., a PE. Intent should be indicated by the notion of a Color as defined in SR Policy Architecture.

  - Signal different instances of a prefix distinguished by color

  - Signal intent associated with a given route

- Support for a flexible NLRI definition to accommodate both efficiency of processing (e.g., packing) and future extensibility

  - Avoid limitations associated with existing SAFI NLRI definitions. For example, 24-bit label.

- Support for validation of paths

  - Reachability of next-hop in control plane

  - Availability and programming of encapsulation in data plane

  - Validation of intent

- Next-hop resolution for Color-Aware route

  - Flexibility to use different intra-domain and inter-domain mechanisms - IGP-FA, SR-TE, RSVP-TE, IGP, BGP-LU etc.

  - Recursive resolution over BGP Color-Aware routes

  - Ability to carry end-to-end cumulative metric for a given color

  - Support setting up an end-to-end Color-Aware path using a different/less preferred or best-effort paths in domains where a particular intent is not available

- Separation of transport and VPN service semantics.

  - Allow for different route distribution planes for service vs transport routes.

- Support signaling of different transport encapsulations

- Support for signaling multiple encapsulations for co-existence and migration

- Generation of BGP Color-Aware routes sourced from IGP-FA, SR-TE policies and BGP-LU from a domain

- Support signaling across domains with different color mappings for a given intent.

8. Future Considerations

Multicast service intent

9. Acknowledgements

Many people contributed to this document.

The authors would especially like to thank Jim Uttaro for his guidance on the work and feedback on many aspects of the problem statement. We would also like to thank Daniel Voyer, Luay Jalil and Robert Raszuk for their review and valuable suggestions.

We also express our appreciation to Bruno Decreane, Keyur Patel, Jim Guichard, Alex Bogdanov, Dirk Steinberg, Hannes Gredler and Xiaohu Hu for discussions on several topics that have helped provide input to

the document. We also thank Huaimo Chen for his valuable review comments.

The authors would like to thank Stephane Litkowski for his detailed review and for making valuable suggestions to improve the quality of the document. We would also like to thank Kamran Raza and Kris Michelson for their review and comments on the document and to Simon Spraggs, Jose Liste and Jiri Chaloupka for their early inputs on the problem statement.

10. References

10.1. Normative References

[I-D.agrawal-spring-srv6-mpls-interworking]
Agrawal, S., Ali, Z., Filsfils, C., Voyer, D., and Z. Li, "SRv6 and MPLS interworking", draft-agrawal-spring-srv6-mpls-interworking-03 (work in progress), August 2020.

[I-D.ietf-bess-srv6-services]
Dawra, G., Filsfils, C., Talaulikar, K., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "SRv6 BGP based Overlay services", draft-ietf-bess-srv6-services-05 (work in progress), November 2020.

[I-D.ietf-idr-bgp-ipv6-rt-constrain]
Patel, K., Raszuk, R., Djernaes, M., Dong, J., and M. Chen, "IPv6 Extensions for Route Target Distribution", draft-ietf-idr-bgp-ipv6-rt-constrain-12 (work in progress), April 2018.

[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-21 (work in progress), January 2021.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-13 (work in progress), October 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-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", draft-ietf-spring-sr-service-programming-03 (work in progress), September 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-28 (work in progress), December 2020.

[I-D.voyer-pim-sr-p2mp-policy]
Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", draft-voyer-pim-sr-p2mp-policy-02 (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>.

[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>.

[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>.

[RFC5512] Mohapatra, P. and E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", RFC 5512, DOI 10.17487/RFC5512, April 2009, <https://www.rfc-editor.org/info/rfc5512>.

[RFC5701] Rekhter, Y., "IPv6 Address Specific BGP Extended Community Attribute", RFC 5701, DOI 10.17487/RFC5701, November 2009, <https://www.rfc-editor.org/info/rfc5701>.

[RFC7311] Mohapatra, P., Fernando, R., Rosen, E., and J. Uttaro, "The Accumulated IGP Metric Attribute for BGP", RFC 7311, DOI 10.17487/RFC7311, August 2014, <https://www.rfc-editor.org/info/rfc7311>.

[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>.

[RFC8277] Rosen, E., "Using BGP to Bind MPLS Labels to Address Prefixes", RFC 8277, DOI 10.17487/RFC8277, October 2017, <https://www.rfc-editor.org/info/rfc8277>.

[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>.

[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>.

10.2. Informative References

[I-D.filsfils-spring-sr-policy-considerations]
Filsfils, C., Talaulikar, K., Krol, P., Horneffer, M., and P. Mattes, "SR Policy Implementation and Deployment Considerations", draft-filsfils-spring-sr-policy-considerations-06 (work in progress), October 2020.

[I-D.ietf-idr-performance-routing]
Xu, X., Hegde, S., Talaulikar, K., Boucadair, M., and C. Jacquenet, "Performance-based BGP Routing Mechanism", draft-ietf-idr-performance-routing-03 (work in progress), December 2020.

[I-D.ietf-mpls-seamless-mpls]
Leymann, N., Decraene, B., Filsfils, C., Konstantynowicz, M., and D. Steinberg, "Seamless MPLS Architecture", draft-ietf-mpls-seamless-mpls-07 (work in progress), June 2014.

[RFC3906] Shen, N. and H. Smit, "Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering Tunnels", RFC 3906, DOI 10.17487/RFC3906, October 2004, <https://www.rfc-editor.org/info/rfc3906>.

[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>.

[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>.

[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>.

[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>.

Authors' Addresses

Dhananjaya Rao
Cisco Systems
USA

Email: dhrao@cisco.com

Swadesh Agrawal
Cisco Systems
USA

Email: swaagraw@cisco.com

Clarence Filsfils
Cisco Systems
Belgium

Email: cfilsfil@cisco.com

Ketan Talaulikar
Cisco Systems
India

Email: ketant@cisco.com

Bruno Decraene
Orange
France

Email: bruno.decraene@orange.com

Dirk Steinberg
Steinberg Consulting
Germany

Email: dws@dirksteinberg.de

Luay Jalil
Verizon
USA

Email: luay.jalil@verizon.com

Jim Guichard
Futurewei
USA

Email: james.n.guichard@futurewei.com

Wim Henderickx
Nokia
Belgium

Email: wim.henderickx@nokia.com

Network Working Group
Internet Draft
Intended status: Standard
Expires: September 8, 2021

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

March 8, 2021

BGP NLRI App Meta Data for 5G Edge Computing Service
draft-dunbar-idr-5g-edge-compute-app-meta-data-02

Abstract

This draft describes a new BGP Network Layer Reachability Information (BGP NLRI) Path Attribute, AppMetaData, for egress router to advertise the running status and environment of the directly attached 5G Edge Computing servers. The AppMetaData can be used by the ingress routers in the 5G Local Data Network to make intelligent path selection for 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, for egress routers to advertise the running status and environment of the directly attached Edge Computing servers. The AppMetaData can be used by the ingress routers in the 5G Local Data Network to make intelligent path selection for 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 the 5G base stations, 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 its 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 decapsulates 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 the 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 time for the SSC [Session and Service Continuity] mode 3 to make the handover process more seamless.

1.2. 5G Edge Computing Network Properties

In this document, 5G Edge Computing Network refers to multiple Local IP Data Networks (LDN) in one region that interconnect the Edge Computing mini-data centers. Those IP LDN networks are the N6 interfaces from 3GPP 5G perspective.

The ingress routers to the 5G Edge Computing Network are the routers directly connected to 5G UPFs. The egress routers to the 5G Edge Computing Network are the routers that have a direct link to the Edge Computing servers. The servers and the egress routers are co-located. Some of those mini Edge Computing Data centers may have Virtual switches or Top of Rack switches between the egress routers and the servers. But transmission delay between the egress routers and the Edge Computing servers is too small to be considered in this document.

When one mini data center has multiple Edge Computing Servers attached to one App Layer Load Balancer, only the App Layer Load Balancer is visible to the 5G Edge Computing Network. How the App Layer Load balancer manages the individual servers is out of the scope of the network layer.

The Edge Computer Services are specially managed services that need to utilize the network topology and balance among multiple mini Edge Computing Data Centers with the same ANYCAST address. UEs can access many services that are not part of the registered 5G Edge Computing Services.

```
   +--+
   |UE|---\+---------+                 +------------------+
   +--+    |  5G     |     +--------+  |   S1: aa08::4450 |
   +--+    | Site +--+-+---+        +----+                |
   |UE|----|  A   |PSA1| 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   |PSA2| Rb |       | R2 | S2: aa08::4460 |
   +--+    |      +--+-+----+       +----+                |
   +--+    |         |  +-----------+  |  S3: aa08::4470  |
   |UE|---/+---------+                 +------------------+
   +--+                                     L-DN2
```

Figure 1: App Servers in different edge DCs

1.3. 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.

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 the ANYCAST address is removing the dependency on UEs. Some UEs (or clients) might use their cached IP addresses instead of querying DNS for an extended period.

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

a) Path Selection: When a new flow comes to an ingress node (Ra), how to select the optimal egress router to reach an ANYCAST server.

   The mechanism described in this draft is for solving this Path Selection problem.

b) How Ingress node keeps the packets from one flow to the same ANYCAST server.

   a.k.a. Flow Affinity, or Flow-based load balancing, which is supported by many commercial routers.

   The ingress node, (Ra/Rb) uses Flow ID (in IPv6 header) or UDP/TCP port number combined with the source address to enforce packets in one flow being placed in one tunnel to one Egress router. No new features are needed.

c) When a UE moves to a new Cell Tower, a method is needed to stick the flow to the same ANYCAST server, which is required by 5G Edge Computing: 3GPP TR 23.748.

This problem is Out of scope for this draft. [5g-edge-compute-sticky-service] describes several approaches to solve this problem.

BGP is an integral part of 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 ingress routers can optimize its path selection algorithm to select an optimal ANYCAST location based on the combination of network delay, the App Server load index, the location capacity index and the location preference.

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

UEs frequent moving from one 5G site to another can make it difficult to plan where the App ANYCAST servers 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.

1.5. 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 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 a 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 hosts 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 an external IP network, in front of a set of application servers. From an IP network perspective, this whole group of servers is 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 the 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. An Edge DC might host 5G core functions in addition to the frequently used application servers.

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. Assumptions

From IP Layer, the Application servers are identified by their IP (ANYCAST) addresses. Here are some assumptions about the 5G Edge Computing services:

- Only the registered Edge Computing services need special consideration in path selection.
- 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 ingress routers' local BGP path compute algorithm includes a special Plugin that can compute the path to the optimal Next Hop (egress router) based on the BGP AppMetaData TLV received for the registered Edge Computing services.

The proposed solution is for the egress 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/bytes 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 an external world. Other data centers can have a 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 the 5G EC environment, the A-ER can be the gateway router to the EC DC where multiple Application servers 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 the 5G Edge Computing environment, the ingress routers to the LDN need 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:

```
               Load-i * CP-j                 Pref-j * Delay-i
Cost-i=min(w *(----------------) + (1-w) *(------------------))
              Load-j * CP-i                 Pref-i * Delay-j
```

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 Site-i, a 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, a 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

The 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.

3.5. Ingress Node BGP Path Selection Behavior

3.5.1. AppMetaData Influenced BGP Path Selection

In this scenario, an ingress router will receive one ANYCAST address's multiple routes from different egress routers that have the direct links to the ANYCAST servers. The ingress router's BGP engine will do path selection, select the best route, and download to FIB. And BGP engine will also download the other paths to FIB that with the AppMetaData taken into the consideration.

Assume that both Ra and Rb in Figure 1 have BGP Multipath enabled. As a result, Dst Address: S1:aa08::4450 is resolved via multiple NextHop: R1, R2, R3.

Suppose the local BGP special Plugin for AppMetaData finds R1 is the best for the flow towards S1:aa08::4450. Then this special Plugin can insert a higher weight for the path R1 so that BGP Best Path is locally influenced by the weight parameter based on the local decision.

3.5.2. Forwarding Behavior

When the ingress router receives a packet and lookup the FIB, get the destination prefix's whole path and AppMetaData. The Forwarding Plane will do computing for the packet and choose the suitable path as the result of the computing. Then the Forwarding Plane encapsulates the packet destined towards the optimal Nexthop node.

For subsequent packets belonging to the same flow, the ingress router needs to forward them to the same egress router unless the selected egress router is no longer reachable. Keeping packets from one flow to the same egress router, a.k.a. Flow Affinity, is supported by many commercial routers.

How Flow Affinity is implemented is out of the scope for this document. Here is one example to illustrate how Flow Affinity can be achieved. This illustration is not to be standardized.

For the registered Edge Computing services, the ingress node keeps a table of

- Service ID (i.e. ANYCAST address)
- Flow-ID
- Sticky Egress ID
- A timer

The Flow-ID in this table is to identify a flow, initialized to NULL. How Flow-ID is constructed is out of the scope for this document. Here is one example of constructing the Flow-ID:

- For IPv6, the Flow-ID can be the Flow-ID extracted from the IPv6 packet header with or without the source address.
- For IPv4, the Flow-ID can be the combination of the Source Address with or without the TCP/UDP Port number.

The Sticky Egress ID is to record the egress node address that the packets of the same flow that have been forwarded to. [5G-Sticky-Service] describes several methods to derive the Sticky Egress ID.

The Timer is always refreshed when a packet with the matching ANYCAST address is received by the node.

If there is no Stick Egress ID present in the table for the ANYCAST address, the forwarding plane computes the optimal path to a NextHop with the AppMetaData taken into consideration. The forwarding plane encapsulate the packet with the tunnel to the chosen NextHop. The chosen NextHop and the Flow ID are recorded in the table entry of the ANYCAST ID.

When the selected optimal egress router is no longer reachable, refer to Section 6 Soft Anchoring on how another path is selected.

3.5.3. Forwarding Behavior after a UE moving to a new 5G Site

When a UE moves to a new 5G Site, the new ingress router might use the pre-computed Egress Router which is passed from the neighboring router. [5G-Edge-Sticky] describes the method for the ingress router connected to the UPF in the new site to take into consideration the information passed from other ingress routers in selecting the optimal egress router. The detailed algorithm is out of the scope of this document.

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 a 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

Raw 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 wi is a value between 0 and 1; w1+ w2+ w3+ w4 = 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. AppMetaData Propagation Scope

AppMetaData is only to be distributed to the relevant ingress nodes of the 5G Edge Computing local data networks. Only the ingress routers that are configured with the 5G Edge Computing services ACLs need to receive the AppMetaData for specific services.

For each registered Edge Computing service, a corresponding filter group can be formed on RR to represent the interested ingress routers that are interested in receiving the corresponding AppMetaData information.

6. 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 a UE to a preferred location of the ANYCAST servers when the preferred location is in good condition. But if there is any failure reaching the preferred location, the Application flow from the UE will be forwarded to another location of the ANYCAST servers.

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 a failure reaching 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 a failure.

Here are the 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 a 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 reply has 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 reply has the ANYCAST address G1, which has the same cost regardless of 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 reply with 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.

7. Manageability Considerations

To be added.

8. Security Considerations

To be added.

9. IANA Considerations

To be added.

10. References

10.1. 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

10.2. 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-Edge-Sticky] 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.

11. 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: September 5, 2021

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

March 7, 2021

BGP UPDATE for SDWAN Edge Discovery
draft-dunbar-idr-sdwan-edge-discovery-04

Abstract

The document describes the 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

Copyright Notice

Table of Contents

1. Introduction

[SDWAN-BGP-USAGE] illustrates how BGP is used as control plane for a SDWAN network. SDWAN network refers to a policy-driven network over multiple different underlay networks to get better WAN bandwidth management, visibility, and control.

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 policy-driven transporting IP packets over multiple different underlay networks to get better WAN bandwidth management, visibility and control.

SDWAN Segmentation: Segmentation is the process of dividing the network into logical sub-networks.

SDWAN VPN: referring to Client's VPN, which is like the VRF on the PEs of a MPLS VPN. One SDWAN client VPN can be mapped one or multiple SD-WAN virtual topologies. How Client VPN is mapped to a SDWAN virtual topology is governed by policies.

SDWAN Virtual Topology: Since SDWAN can connect any nodes, whereas MPLS VPN connects a fixed number of PEs, one SDWAN Virtual Topology refers to a set of edge nodes and the tunnels (including both IPsec tunnels and/or MPLS tunnels) interconnecting those edge nodes.

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 and their associated properties for its attached clients traffic to communicate. The attributes to be propagated includes the SDWAN (client) VPNs supported, the attached routes under specific SDWAN VPNs, 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 VPNs can only be exchanged to the SDWAN peer nodes that are authorized to communicate.

3.2. Comparing with Pure IPsec VPN

Pure IPsec VPN has IPsec tunnels connecting all edge nodes via public internet, therefore requires stringent authentication and authorization (i.e. IKE Phase 1) before other properties of IPsec SA can be exchanged. The IPsec Security Association (SA) between two untrusted nodes typically requires the following configurations and message exchanges:

```
    - IPsec IKE to authenticate with each other
    - Establish IPsec SA
         o Local key configuration
         o Remote Peer address (192.10.0.10<->172.0.01)
         o IKEv2 Proposal directly sent to peer
            o Encryption method, Integrity sha512
         o Transform set
    - Attached client prefixes discovery
         o By running routing protocol within each IPsec SA
         o If multiple IPsec SAs between two peer nodes are
            established to achieve load sharing, each IPsec tunnel
            needs to run its own routing protocol to exchange client
            routes attached to the edges.
    - Access List or Traffic Selector)
         o Permit Local-IP1, Remote-IP2
```

In a BGP controlled SDWAN network, e.g. a MPLS based network adding short-term capacity over Internet using IPsec, there are secure connection between edge nodes and RR, via private path, TLS, DTLS, etc. The authentication of peer nodes is managed by the RR. More importantly, when an edge node needs to establish multiple IPsec tunnels to many different 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 authentication in a BGP Controlled SDWAN network.

Client VPNs are configured via VRFs, just like the configuration of the existing MPLS VPN. The IPsec equivalent traffic selectors for local and remote routes is achieved by importing/exporting VPN Route Targets. The binding of client routes to IPsec SA is dictated by policies. As the result, the IPsec configuration for a BGP controlled SDWAN (with mixed MPLS VPN) can be simplified as the following:

- SDWAN controller has authority to authenticate edges and peers. Remote Peer association is controlled by the SDWAN Controller (RR)
- The IKEv2 proposals including the IPsec Transform set can be sent directly to Peer or incorporated with BGP UPDATE.
- BGP UPDATE: Announce the client route reachability for all permitted parallel tunnels/paths.
  - o No need to run multiple routing protocols in each IPsec tunnel.
- using importing/exporting Route Targets under each client VPN (VRF) to achieve the traffic selection (or permission) among clients' routes attached to multiple edge nodes.

3.3. Client Route UPDATE and Hybrid Underlay Tunnel UPDATE

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

- UPDATE U1 for advertising the attached client routes, This UPDATE is exactly the same as the BGP edge client route UDPDATE. It uses the Encapsulation Extended Community and the Color Extended Community to link with the Underlay Tunnels UPDATE Message 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 advertises the properties of the various tunnels, including IPsec, terminated at the edge node. This UPDATE is for an edge node to advertise the properties of directly attached underlay networks, including the NAT information, pre-configured IPsec SA identifiers, and/or the underlay network ISP information. This UPDATE can also include the detailed IPsec SA attributes, such as keys, nonce, encryption algorithms, etc.

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 VPN 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-Hybrid
     (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 for 192.0.0.1 [Tunnel-encap]
      IPsec-SA-ID sub-TLV [See the Section 6]
    Tunnel encapsulation Path Attribute [type=SDWAN-Hybrid]
      IPSec SA for
      Tunnel-End-Point Sub-TLV /* for 170.0.0.1 */
      IPsec-SA-ID 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.

3.4. Edge Node Discovery

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

- Secure connection to a SDWAN controller (i.e. RR in this context):
  For a SDWAN edge with both MPLS and IPsec path, the edge node should already have secure connection to its controller, i.e. RR in this context. For a remote SDWAN edge that is only accessible via Internet, the SDWAN edge node, upon power up, establishes 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 connection.

- 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

4.1. SDWAN Segmentation, SDWAN Virtual Topology and Client VPN

In SDWAN deployment, "SDWAN Segmentation" is a frequently used term, referring to partitioning a network to multiple sub-networks, just like what MPLS VPN does. "SDWAN Segmentation" is achieved by creating SDWAN virtual topologies and SDWAN VPNs. A SDWAN virtual topology consists of a set of edge nodes and the tunnels, including both IPsec tunnels and/or MPLS VPN tunnels, interconnecting those edge nodes.

A SDWAN VPN is same as a client VPN, which is configured in the same way as the VRFs on PEs of a MPLS VPN. One SDWAN client VPN can be mapped to one or multiple SD-WAN virtual topologies. How a Client VPN is mapped to a SDWAN virtual topology is governed by policies from the SDWAN controller.

Each SDWAN edge node may need to support multiple VPNs. Just like Route Target is used to distinguish different MPLS VPNs, SDWAN VPN ID is used to differentiate the SDWAN VPNs. For example, in the picture below, the "Payment-Flow" on C-PE2 is only mapped to the virtual topology of C-PEs to/from Payment Gateway, whereas other flows can be mapped to a multipoint to multipoint virtual topology.

```
                                          +---+
                           +--------------|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 Virtual Topology & VPN

4.2. 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 VPN #1                                          SDWAN VPN #1
     SDWAN VPN #2                                          SDWAN VPN #3
```

Figure 3: Constraint propagation of Edge Property

However, a SDWAN overlay network can span across untrusted networks, RR can't trust the RT Constraint (RTC) NLRI BGP UPDATE from any nodes. RR can only process the RTC NLRI from authorized peers for a SDWAN VPN.

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 VPNs.

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 VPN.

5. Client Route UPDATE

The SDWAN network's Client Route UPDATE message is same as the MPLS VPN client route UDPATE message. The SDWAN Client Route UPDATE message uses the Encapsulation Extended Community and the Color Extended Community to link with the Underlay Tunnels UPDATE Message.

5.1. SDWAN VPN ID in Client Route Update

SDWAN VPN is same as client VPN in BGP controlled SDWAN network. The Route Target Extended Community should be included in a Client Route UPDATE message to differentiate the client routes from routes belonging to other VPNs.

5.2. SDWAN VPN ID in Data Plane

For a SDWAN edge node which can be reached by both MPLS and IPsec paths, 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 VPN ID. For NVO (VxLAN, GENEVE, etc.) encapsulation within the IPsec tunnel, Virtual Network Identifier (VNI) field is used to carry the SDWAN VPN ID.

6. Hybrid Underlay Tunnel UPDATE

The hybrid underlay tunnel UPDATE is to advertise the detailed properties of hybrid types of tunnels terminated at a SDWAN edge node.

A client route UDPATE is recursively tied to an underlay tunnel UDPATE by the Color Extended Community included in client route UPDATE.

6.1. NLRI for Hybrid Underlay Tunnel Update

A new NLRI is introduced within the MP_REACH_NLRI Path Attribute of RFC4760, for advertising the detailed properties of hybrid types of tunnels terminated at the edge node, with SAFI=SDWAN (code = 74):

```
     +-------------------+
     |   NLRI Length     | 1 octet
     +-------------------+
     |   Site-Type       | 2 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: 2 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 a simple deployment, such as all edge nodes under one SDWAN management system, the node ID is enough for the SDWAN management to map the site to its precise geolocation.
  Site-Type = 2: For large SDWAN heterogeneous deployment where a Geo-Loc Sub-TLV [LISP-GEOLoc]is needed to fully describe the accurate location of the node.
- Port local ID: SDWAN edge node Port identifier, which is locally significant. If the SDWAN NLRI applies to multiple ports, this field is NULL.
- SDWAN-Color: to correlate with the Color-Extended-community included in the client routes UPDATE.
- SDWAN Edge Node ID: The node's IPv4 or IPv6 address.

6.2. 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
```

6.3. IPsec-SA-ID Sub-TLV

IPsec-SA-ID Sub-TLV is for the Hybrid Underlay Tunnel UPDATE to reference one or more preestablished IPsec SAs by using their identifiers, instead of listing all the detailed attributes of the IPsec SAs.

Using IPsec-SA-ID Sub-TLV not only greatly reduces the size of BGP UPDATE messages, but also allows the pairwise IPsec rekeying process to be performed independently.

The following is the structure of the IPsec-SA-ID sub-TLV:

```
    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= IPsec-SA-ID subTLV      |  Length (2 Octets)            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      IPsec SA Identifier #1                   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      IPsec SA Identifier #2                   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 prepended right before the IPsec-SA-ID Sub-TLV.

6.3.1. Encoding example #1 of using IPsec-SA-ID Sub-TLV

This section provides an encoding example for the following scenario:

- There are four IPsec SAs terminated 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
- two 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                           |
   ~                                                               ~

   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                      GRE Sub-TLV                              ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | subTLV-Type                     = IPsec-SA-ID       |      Length =
       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     IPsec SA Identifier = 1                   |
   +---------------------------------------------------------------+
   |                     IPsec SA Identifier = 2                   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                      VxLAN Sub-TLV                            ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |subTLV-Type = IPsec-SA-ID      |        Length=                |
   +-------------------------------+-------------------------------+
   |                     IPsec SA Identifier = 3                   |
   +-------------------------------+-------------------------------+
   |                     IPsec SA Identifier = 4                   |
   +---------------------------------------------------------------+
```

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

- Tunnel-end-point sub-TLV total length
- The GRE Sub-TLV total length,
- The IPsec-SA-ID Sub-TLV length,
- The VxLAN sub-TLV total length, and
- The IPsec-SA-ID Sub-TLV length.

6.3.2. Encoding Example #2 of using IPsec-SA-ID Sub-TLV

For IPsec SAs terminated at different endpoints, multiple Tunnel Encap Attributes must be included. This section provides an encoding example for the following scenario:

- there is one IPsec SA terminated at the WAN Port address 192.0.0.1; and another IPsec SA terminated 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                                     ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                   GRE Sub-TLV                                 ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~           IPsec-SA-ID sub-TLV #1                              ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Tunnel-Type =SDWAN-Hybrid    |       Length =                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            Tunnel-end-Point Sub-TLV                           |
   ~            for  170.0.0.1                                    ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                   GRE sub-TLV                                 ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~             IPsec-SA-ID sub-TLV #2                            ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

6.4. 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.

o 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).
o 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)
o Transport Network ID.Central Controller assign a global unique ID to each transport network.
o RD ID.Routing Domain ID.need to be global unique.
o Local IP.The local (or private) IP address of the port.
o Local Port.used by Remote SDWAN edge node for establishing IPsec to this specific port.
o Public IP.The IP address after the NAT. If NAT is not used, this field is set to NULL.
o Public Port.The Port after the NAT. If NAT is not used, this field is set to NULL.

6.5. 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 SA Property Sub-TLVs

This section describes the detailed IPsec SA properties sub-TLVs.

7.1. 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 SDWAN VPN ID 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 in Section 6. 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.

7.2. 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                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

7.3. 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.

7.4. 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.

7.5. 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 Hybrid (SDWAN) Overlay SAFI = 74 assigned by IANA
- o IPsec-SA-ID Sub-TLV 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 Yoav Nir, Graham Bartlett, Jim Guichard, John Scudder, and Donald Eastlake for their review and discussions.

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

Inter-Domain Routing
Internet-Draft
Updates: 8955 (if approved)
Intended status: Standards Track
Expires: 11 October 2021

J. Haas
Juniper Networks
9 April 2021

BGP Flowspec Capability Bits
draft-haas-flowspec-capability-bits-02

Abstract

BGP Flowspec (RFC 8955) provides the ability to filter traffic using various matching components. The NLRI format currently defined does not permit incremental deployment of new BGP Flowspec components. This draft defines a new BGP Capability to permit incremental deployment of such new Flowspec component types.

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 BCP 14 [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

BGP Flowspec [RFC8955] provides a mechanism to distribute traffic flow spcifications into BGP. One general purpose of these flow specifications is for the distribution of firewall rules to receiving routers, particularly for mitigating distributed denial of service (DDoS) attacks. The flow specification rules are encoded as BGP NLRI [RFC4271].

The matching components of a flow specification NLRI is a serialized set of optional components. The components are documented in [RFC8955], Section 3. [RFC8956] defines IPv6-specific components. The full set of Flowspec component types is maintained in an IANA registry located at the IANA Flow Spec Component Types registry (https://www.iana.org/assignments/flow-spec/flow-spec.xhtml).

Unknown Flowspec component types require treatment as a malformed NLRI ([RFC8955], Section 4.2). This is due to the lack of a mandatory length element for the components in the NLRI. Without such a length, it is not possible to determine how to properly decode unknown components in the Flowspec NLRI.

There has been active interest in the IDR Working Group to extend BGP Flowspec for additional purposes. However, with this difficulty in being able to handle unknown components, those new features are unable to be deployed in a BGP Flowspec domain in an incremental fashion. Either a carefully managed "flag day" deployment is required to avoid disrupting existing sessions, or the Flowspec domain is carefully managed such that devices with incompatible sets of known/unknown components are carefully separated in a "ships in the night" scenario. Both options are fragile and operationally cumbersome.

Some initial discussion has begun for a version 2 of Flowspec in [I-D.hares-idr-flowspec-v2]. That document may eventually address this incremental deployment issue, along with a number of other items.

This document proposes to address the issues of incremental deployment of new BGP Flowspec component types via a new BGP Capability [RFC5492], the BGP Flowpec Capability Bits.

2. BGP Flowspec Capability Bits

BGP Flowspec component types are one octet in length with values in the range from 0..255. The BGP Flowspec Capability Bits encode a bit-string where each supported component type has its respective bit set when the BGP Speaker is willing to receive BGP Flowspec NLRI that contain that component type.

The BGP Flowspec Capability Bits Capability is encoded as follows:

* Capability Code of (TBD).

* Capability Length of 1..32.

* Capability Value contains a bit-string where a bit is set if the underlying BGP Flowspec component is willing to be accepted by BGP Speaker advertising this capability.

```
               Example encoding for Capability Value:

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

Bit 0 set to 0, bits 1..13 set to 1 showing support for all capabilities for IPv6 Flowspec, bits 14 and 15 are set to 0.

3. Operation

BGP Flowspec Capability Bits not advertised in the encoded bit-string are treated as if they were sent with a value of zero for that bit.

The Capability Length reflects the number of octets it takes to encode the BGP Flowspec Capability Bits. While the total number of octets required to represent the entire range of component types is only 32 octets, implementations SHOULD limit the number of octets transmitted to those required to encode the final one-bit. Space in BGP Capabilities may be limited in some implementations depending on the number of capabilities to be sent. (See [I-D.ietf-idr-ext-opt-param] for discussion on a feature to address this point.)

Bit-values 0 and 255 SHOULD be set to zero as they are RESERVED.

The BGP Flowspec Capability Bits Capability SHOULD be sent by a BGP Speaker utilizing any AFI/SAFI using BGP Flowspec encoding as defined in [RFC8955], or [RFC8956].

The BGP Flowspec Capability Bits Capability MUST be sent by a BGP Speaker utilizing BGP Flowspec encoding with a component type not defined in those documents previously mentioned. (I.e. component types not in the range 1..13.)

A BGP Speaker that has received the BGP Flowspec Capability Bits Capability MUST NOT originate or propagate a BGP Flowspec encoded NLRI that contains a component types that is not present in the received bit-string.

A BGP Speaker that has received a BGP Flowspec related AFI/SAFI without this Capability MUST treat the absence as equivalent to having received the Capability Bits covered by the base specification for its defining RFC, [RFC8955] or [RFC8956].

4. Propagation of Known Components and Mismatch with Local Filtering Capabilities

There may be circumstances where a BGP Speaker is capable of parsing Flowspec components that it is not capable of implementing as filters. Section 4.2 of [RFC8955] specifies that:

> "All combinations of components within a single Flow Specification are allowed. However, some combinations cannot match any packets (e.g., "ICMP Type AND Port" will never match any packets) and thus SHOULD NOT be propagated by BGP."

This document updates that text to:

> "All combinations of components within a single Flow Specification are allowed. However, some combinations cannot match any packets (e.g., "ICMP Type AND Port" will never match any packets) and thus SHOULD NOT be propagated by BGP.
>
> "When BGP Flowspec component types are understood and the operator determines that deployment-wide filtering intent would not be comporomised by propagating Flowepec routes that cannot match any packets, it SHOULD propagate the route in BGP. This permits NLRI with known components to be propagated to downstream BGP Speakers in the deployment."

5. BGP Flowspec Implications for Filtered NLRI

BGP Flowspec NLRI encode match operations for traffic filtering rules. Filtering is an ordered operation. Since the current encoding of the NLRI does not supply explicit filtering order, the protocol imposes a forwarding order based on the contents of the NLRI.

When a BGP Flowspec NLRI is not propagated due to filtering by this feature, or by user policy, there is the potential that the network-wide filtering intent may be compromised by the missing rules. The exact impact of this on filtering will depend on the relative independence of the full set of BGP Flowspec routes in the BGP Flowspec routing domain.

Operators must exercise care when deploying BGP Flowspec features with new component types to understand the propagation of such routes in their deployment, and the impact that filtering may have on the routes they wish to originate.

6. Error Handling

If a BGP Speaker implementing this document has transmitted BGP Flowspec Capability Bits to its peer and receives a BGP Flowspec NLRI with an unacceptable component (not in its bit-string), it MAY terminate the BGP session by sending a NOTIFICATION message.

7. Acknowledgements

Thanks to Aseem Choudhary, Jakob Heitz, Christoph Loibl, Robert Raszuk for their comments on this proposal.

8. Security Considerations

All of the Security Considerations for [RFC8955] and [RFC8956] still apply.

Additionally, the BGP Flowspec Capability Bits may cause implicit filtering of some BGP Flowspec NLRI in a Flowspec domain. Depending on the relative independence of the traffic matched by the BGP Flowspec rules in the ordering required by their specifications, such filtered NLRI may result in impact to the desired domain-wide filtering behaviors.

9. IANA Considerations

IANA is requested to assign a new BGP Capability to the Capability Codes registry from the First Come, First Served pool. The Reference for the registration is this document. The Change Controller is IETF.

10. References

10.1. 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>.

[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>.

[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>.

[RFC8955] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", RFC 8955, DOI 10.17487/RFC8955, December 2020, <https://www.rfc-editor.org/info/rfc8955>.

[RFC8956] Loibl, C., Ed., Raszuk, R., Ed., and S. Hares, Ed., "Dissemination of Flow Specification Rules for IPv6", RFC 8956, DOI 10.17487/RFC8956, December 2020, <https://www.rfc-editor.org/info/rfc8956>.

10.2. Informative References

[I-D.hares-idr-flowspec-v2]
Hares, S., "BGP Flow Specification Version 2", Work in Progress, Internet-Draft, draft-hares-idr-flowspec-v2-00, 25 June 2016, <http://www.ietf.org/internet-drafts/draft-hares-idr-flowspec-v2-00.txt>.

[I-D.ietf-idr-ext-opt-param]
Chen, E. and J. Scudder, "Extended Optional Parameters Length for BGP OPEN Message", Work in Progress, Internet-Draft, draft-ietf-idr-ext-opt-param-09, 21 August 2020, <http://www.ietf.org/internet-drafts/draft-ietf-idr-ext-opt-param-09.txt>.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, DOI 10.17487/RFC2578, April 1999, <https://www.rfc-editor.org/info/rfc2578>.

[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>.

Appendix A. Encoding of the Bit-String

IETF has a mixed history in terms of how bit numbering is described. The format as used in this document, where the left-most bit sent on the wire is bit zero, is consistent with IETF PDU diagrams and also the SNMP BITS construct [RFC2578], Section 7.1.4.

That said, the author is aware of how annoying the code for that construct can be.

Appendix B. Open Issues

* Are there circumstances where advertising capability bits for BGP Flowspec NLRI that need to vary on a per AFI-SAFI basis? Currently, the IANA registry is a single name space for all supported and proposed BGP address families. As an example, the Flowspec for NVO3 feature has components that are defined that do not have incremental deployment issues due to being well formed with a length field. However, since it still includes existing Flowspec filtering for the outer and inner IP headers, the issues addressed by this proposal still apply.

Author’s Address

Jeffrey Haas
Juniper Networks

Email: jhaas@juniper.net

Network Working Group R. Bush
Internet-Draft Arrcus, Inc. & Internet Initiative Japan
Intended status: Informational J. Dong
Expires: September 9, 2021 Huawei Technologies
J. Haas, Ed.
Juniper Networks
W. Kumari, Ed.
Google
March 8, 2021

Requirements and Considerations in BGP Peer Auto-Configuration
draft-ietf-idr-bgp-autoconf-considerations-00

Abstract

This draft is an exploration of the requirements, the alternatives, and trade-offs in BGP peer auto-discovery at various layers in the stack. It is based on discussions in the IDR Working Group BGP Autoconf Design Team. The current target environment is the datacenter.

This document is not intended to become an RFC.

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 BCP 14 [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

This draft is an exploration of the requirements, the alternatives, and trade-offs in BGP peer auto-discovery at various layers in the stack. It is based on discussions in the IDR Working Group BGP Autoconf Design Team. The current target environment is the datacenter.

2. Design Team Determinations

2.1. Problem Scope

The current target environment is BGP as used for the underlay routing protocol in data center networks. Other scenarios may be considered as part of the analysis for this work, but work on those environments will be deferred to other efforts.

2.2. Simplicity

The auto-discovery mechanism is designed to be simple.

The goal is to select BGP Speakers where a BGP session may be successfully negotiated for a particular purpose. The auto-discovery mechanism will not replace or conflict with data exchanged by the BGP FSM, including its OPEN message.

2.3. BGP Auto-Discovery Protocol State Requirements

Tersely, the required state that MUST be carried by the BGP Auto-Discovery Protocol for a discovered session include:

BGP Session Transport State:

- IP addresses
- Transport security parameters
- GTSM [RFC5082] configuration, if any
- BFD [RFC5880] configuration, if any

BGP Session Protocol State:

- AS Numbers
- BGP Identifier
- Supported AFI/SAFIs
- Device Role

Once this information has been learned, discovery has been completed. The BGP Speaker has the necessary information to determine if it wishes to open a BGP session with the discovered BGP Speaker. It can then then initiate a BGP session with the discovered BGP Speaker.

2.3.1. BGP Session Transport State

- Support for IPv4 and IPv6 address families, but do not assume that both are available.
- The ability to use directly attached interface addresses, or the device’s Loopback address. When using the Loopback address, potentially exchange additional information to bootstrap forwarding to that address.
- Discovery of BGP transport protocol end-points and essential properties such as IP addresses, transport security parameters, layer 3 liveness mechanisms such as BFD, and support for GTSM.
- Transport security parameters include protocol - such as plain TCP, TCP-AO [RFC5925], IPsec [RFC4301], TCP-MD5 [RFC2385] - and necessary configuration for that protocol. Some example considerations for this are represented in YANG Data Model for Key Chains [RFC8177].

2.3.2. BGP Session Protocol State

- Discovery of BGP peer session parameters relevant to peer selection such as Autonomous System (AS) Numbers, BGP Identifiers, supported address families/subsequent-address families (AFI/SAFIs), and device roles.

2.3.2.1. BGP Auto-Discovery Device Role Requirements

As part of peer selection, it may be necessary to understand the role of the discovered session to determine whether or not the BGP Speaker desires to establish a peering session.

In some cases, role may be a clear function of the device in a deployment. An example of this is a discovered session for a BGP Clos fabric: The interface may for a leaf, the aggregation layer, or the spine layer. Even then, the type of fabric, what pod it belongs to and the peer type may be relevant information.

Some types of device roles may be subject to standardization, such as BGP Clos fabrics. Flexibility to permit operators to use device roles for their own auto-configuration peer selection purposes is a requirement.

Peering sessions may need to advertise that they are capable to be used for multiple roles. Thus, it is a requirement that the auto-discovery protocols permit advertising multiple roles in the same PDU.

2.4. BGP Auto-Discovery Protocol Transport Requirements

BGP Auto-Discovery Protocol State may be carried in multiple protocols operating in different transport layers.

Implementations supporting more than one protocol for this state must have a mechanism for consistently selecting discovered BGP sessions. The BGP Identifier, which is carried by the BGP OPEN message, can help detect sessions to the same BGP Speaker carried in multiple protocols.

2.5. Operator Configuration

With BGP auto-discovery, some configuration of BGP is still needed. Operator configuration should be able to decide at least the following:

- o Select or otherwise filter which peers to actually try to send BGP OPEN messages.
    - * Permit the matching of device role from the discovery protocol as part of peer selection.
- o Decide the parameters to use. For example:
    - * IP addressing: IPv4 or IPv6.
    - * Interface for peering: Loopback, or Direct.
    - * Any special forwarding or routing needed for reaching the prospective peer; for example, loopback.
    - * AS numbering.
    - * BGP Transport Security Parameters.
    - * BGP Policy that is appropriate for the type of discovered session.

In addition to actually forming the BGP sessions, a common deployment model may also be the so called "validation" model. In this model, the operator configures the BGP sessions manually, and uses the information collected/populated by the BGP Autoconf mechanism to validate that the sessions are correct.

3. Design Principle Considerations

This section summarizes the considerations of possible criteria for the design of a BGP auto-discovery mechanism, which may need further discussion in a wider community than the design team; for example, the IDR Working Group.

3.1. Transport Considerations

The network layer of the discovery mechanism may impact the scoping of the deployment of the auto-discovery mechanism.

- Layer 2: For example, based on Ethernet.
- Layer 3: Which is generic for any link-layer protocol.

Potentially leveraging existing protocols deployed in the data center.

The length of messages supported by the protocol.

How extensible the protocol is to carry future state for BGP auto-configuration.

3.2. Auto-Discovery Protocol Timing Considerations

Establishing a reasonable expectation for the timeliness of auto-configuration is desirable. When a link is plugged-in, one shouldn’t have to wait minutes for potential peers to be discovered and BGP session establishment attempted. For protocols crafted explicitly for BGP auto-configuration, the time for discovery should be a reasonable amount of time; for example ten seconds or less.

Since discovery mechanisms may become very chatty when utilized by a number of devices on shared networks, the protocol should not impose undue burden on the devices on that network to process the discovery messages. New auto-discovery protcols MUST NOT transmit messages more than once a second.

When an auto-discovery mechanism is used for a point-to-point link, or with the expectation of establishing a BGP session with a single BGP Speaker on that network, the auto-discovery protocol MAY quiesce once the discovered BGP session has become Established.

In cases where the auto-discovery protocol is carried as state in another protocol, that protocol will have its own timeliness considerations. The auto-discovery mechanism SHOULD NOT interfere with the timing of the existing protocol.

3.3. Relationship with BGP

- The auto-discovery mechanism should be independent from BGP session establishment.
- Not affect on BGP session establishment and routing exchange, other than the interactions for triggering the setup/removal of peer sessions based on the discovery mechanism.
- Potentially leveraging existing BGP protocol sessions for discovery of new BGP sessions.

3.4. Session Selection Considerations

Candidate BGP sessions to a given BGP Speaker may be discovered by one or more auto-discovery protocols. Even for a single protocol, multiple transport session endpoints may be discovered for the same BGP Speaker. These different sessions may be required for supporting different address families, such as IPv4/IPv6, depending on the BGP operational practices for that device. Examples include a distinct and matching session for the IPv4/IPv6 address family, a unified session carrying IPv4 over IPv6 and vice-versa, etc.

The BGP Identifier (router-id), a required protocol component of BGP, can serve to identify the same instance of the BGP Speaker. This is a required element of the information to be carried in the auto-discovery protocol.

When multiple mechanisms exist to discovery the same BGP speaker in an implementation, that implementation MUST document the process by which it chooses discovered peers. Those implementations also MUST describe interactions with their protocol state machinery for each mechanism.

3.5. Operational Trust Considerations

Different deployment models will have different trust models and requirements. Some of this will be driven by the size, complexity and operational practices of the operator. For example, some operators have very strict physical protection of the datacenter, and their deployment model assumes that anything which plugs into devices in the datacenter is, by definition, trusted. Other operators take a very different approach, and assume the least possible amount of trust.

Much of this difference is also reflected in the operator’s bootstrapping solution. Some operators build individual configurations for each device, and manually provision the configuration into the non-volatile storage of the device before it is shipped. Other operators use solutions similar to PXE Boot to automatically load an operating system and configuration onto the device, based on a unique device identifier (such as management Ethernet MAC address). Some operators pre-configure devices with identical base configurations containing some bootstrapping policy logic (e.g., "If you are a Model-X device, and interface 23 is connected to a device of type Y, then you must be at Stage-2 in a Clos fabric") and allow the device to use this policy information to infer its role and position. A final set of datacenter operators, for example enterprises, would like to be able to simply unpack a new device, plug it in and have the device infer everything. (It is unclear if this is a deployment model that we want to support.)

Many datacenter operators already have a well-developed process for installing and bringing up a new datacenter network, complete with solutions to bootstrap and configure the network. These operators will want to be able to use the BGP Autoconf mechanism to perform validation of the datacenter fabric, and ongoing "sanity-checking" to confirm that the datacenter is correctly cabled, and that the BGP sessions which have been configured from the database match what the autodiscovered sessions would have created. Over time, if the BGP Autoconf solution proves to be successful, reliable, and scaleable, operators may begin using it as the primary source of record.

Closely related to these considerations is the "scope" of the discovery process. It is expected that many operators will wish to only perform discovery on "infrastructure" or "fabric" interfaces, and not interfaces which face customers.

It is not clear that the solution that chosen will be able to meet all of the trust and deployment models, and we will need to prioritize which set(s) of deployment scenarios are the most important for the Working Group to solve.

Trust/Operational deployment driven requirements. The solution should:

- Allow operators to determine which classes of interfaces the discovery protocol operates on (e.g: "Interfaces numbered 1-17" or "Only 100GE interfaces"). This is likely an implementation detail.
- Allow operation in a "validation" or "verification" only mode, where the Autoconf solution populates a database or model showing what sessions it would bring up if allowed.
- Ideally allow for different levels of "granularity" in pre-configuration. For example, if the protocol is capable of autoconfiguring everything, it should also support filtering or limiting the session according to configured policy. (Likely an implementation detail.)

- o Support preconfigured authentication systems. This is an area where more discussion is needed! The solution MUST also support a "no authentication" mode. Negotiated keying solutions, such as IKE, may be desireable but not mandatory for the solution.
- o Support Ethernet sub-interfaces such as VLANs.
- o Support non-Ethernet interfaces. This may include tunnels.

3.6. Error Handling Considerations

The purpose of the BGP auto-discovery protocol is to discover potential BGP sessions and provide enough information for a BGP Speaker to start a BGP session. It is possible for the information present in the auto-discovery protocol to not match the session's information. Such mis-matches will result in different classes of problems:

- o The BGP transport session may not connect. This could be the result of mismatches in IP addresses, GTSM configuration, BGP transport security configuration, etc. In these cases, a BGP Speaker attempts to establish a session and fails. Implementations SHOULD provide a way to clear such discovered sessions or exclude them from further connect attempts.
- o The BGP transport session connects, but the parameters in the BGP OPEN message do not match those in the auto-discovery protocol. In this case, the implementation may wish to disconnect from the BGP session and exclude it from further connection attempts. The implementation SHOULD raise a visible fault to the operator. The implementation SHOULD provide a mechanism to permit further attempts to connect to the discovered session.
- o The operator may choose to leverage the auto-discovery mode for validation purposes only. The implementation should provide access to the operator for discovered BGP sessions from the auto-discovery protocol; for example via the user-interface. The implementation SHOULD permit a manually configured BGP session to conflict with information present in the auto-discovery protocol, but SHOULD raise an alarm with the operator that this has been done.

4. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

5. Security Considerations

There are two primary components to be secured in environments utilizing BGP auto-discovery: The BGP transport layer discovered via the protocol, and the auto-discovery protocol itself.

5.1. BGP Transport Security Considerations

The purpose of the auto-discovery protocol is to ease the setup of BGP sessions for various applications, including data-center fabrics. However, care must be taken to not permit sessions to be setup outside of trusted environments. It is RECOMMENDED that sessions advertised using BGP auto-discovery be protected at the transport layer using mechanisms such as TCP-AO, IPsec, or the deprecated TCP-MD5.

It is thus a requirement that the auto-discovery protocol carry sufficient information to determine what transport security is to be used when establishing a BGP session.

All Security Considerations from [RFC4272], BGP Security Vulnerabilities Analysis, continue to apply.

5.2. Auto-discovery Protocol Considerations

As noted in previous sections, BGP auto-discovery be scoped to different portions of the network dependent on the network layar at which it is deployed. The information present in the auto-discovery protocol is considered sensitive, since it identifies resources running the BGP protocol. Care should be exercised in avoiding inadvertent disclosure of BGP sessions that are configured to permit auto-configuration even when BGP session transport security is in use. The auto-discovery protocol sets the context for such inadvertent disclosure.

5.2.1. Potential Scopes of an Auto-discovery Protocol

A Layer 2 unicast protocol targets a known device, potentially discovered through other means. The targeted device receives the message. Depending on the Layer 2 environment, other devices on the same link may may be able to observe the protocol messages. Point to point links may also fall into this category.

A Layer 2 multicast protocol targets a group of devices on that Layer 2 multicast domain. A set of devices in that domain receives the message. Such messages may cross a number of devices in the domain. An example of this includes a set of Ethernet switches.

A Layer 3 unicast protocol inherits the properties of the Layer 2 protocol, and is intended to address a specific address - typically one device. Layer 3 unicast protocols may leverage GTSM for their security.

A Layer 3 multicast protocol addresses a group of devices in a given multicast domain. Such domains may be scoped, such as a single link's "All-Routers" group or perhaps all devices subscribed to a specific multicast group in a network. In many cases, a Layer 3 multicast protocol inherits the properties of the Layer 2 multicast protocol. Link-local scoped multicast protocols may be able to leverage GTSM.

A Layer 7 protocol is scoped per the mechanism in the underlying protocol. IGPs such as OSPF and IS-IS provide an "internal" scoping. BGP, depending on the deployment of the underlying address family, may vary from a targeted advertisement, to Internet-wide.

Each of these scopes provide different opportunities for inadvertent disclosure. The auto-discovery protocol will need to address how the desired security properties interact with the protocol scope.

5.2.2. Desired Security Properties of the Auto-discovery Protocols

Data Integrity is a required property. The data that is transmitted by a speaker of the auto-configuration protocol should be able to pass among its speakers properly.

Peer Entity authentication is a required property for Layer 2 and Layer 3 implementations. In a Layer 7 protocol, that protocol may provide the necessary authentication.

Confidentiality is an optional property. There is a tension between the desire to provide for a simple auto-configuration protocol that is easy to diagnose and debug with inadvertent disclosure.

The auto-configuration protocol must be resistant to Denial of Service, and to causing Denial of Service to discovered BGP session end-points.

6. Acknowledgments

The IDR BGP Auto-Conf Design Team.

7. References

7.1. 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>.

[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>.

7.2. Informative References

[I-D.acee-idr-lldp-peer-discovery] Lindem, A., Patel, K., Zandi, S., Haas, J., and X. Xu, "BGP Logical Link Discovery Protocol (LLDP) Peer Discovery", draft-acee-idr-lldp-peer-discovery-08 (work in progress), December 2020.

[I-D.acee-ospf-bgp-rr] Lindem, A., Patel, K., Zandi, S., and R. Raszuk, "OSPF Extensions for Advertising/Signaling BGP Route Reflector Information", draft-acee-ospf-bgp-rr-01 (work in progress), September 2017.

[I-D.ietf-lsvr-l3dl] Bush, R., Austein, R., and K. Patel, "Layer 3 Discovery and Liveness", draft-ietf-lsvr-l3dl-07 (work in progress), January 2021.

[I-D.ietf-lsvr-l3dl-signing] Bush, R. and R. Austein, "Layer 3 Discovery and Liveness Signing", draft-ietf-lsvr-l3dl-signing-01 (work in progress), January 2021.

[I-D.ietf-lsvr-l3dl-ulpc] Bush, R. and K. Patel, "L3DL Upper Layer Protocol Configuration", draft-ietf-lsvr-l3dl-ulpc-01 (work in progress), January 2021.

[I-D.ietf-lsvr-lsoe] Bush, R., Austein, R., and K. Patel, "Link State Over Ethernet", draft-ietf-lsvr-lsoe-01 (work in progress), February 2019.

[I-D.raszuk-idr-bgp-auto-discovery]
Raszuk, R., Mitchell, J., Kumari, W., Patel, K., and J. Scudder, "BGP Auto Discovery", draft-raszuk-idr-bgp-auto-discovery-06 (work in progress), December 2019.

[I-D.raszuk-idr-bgp-auto-session-setup]
Raszuk, R., "BGP Automated Session Setup", draft-raszuk-idr-bgp-auto-session-setup-01 (work in progress), December 2019.

[I-D.xu-idr-neighbor-autodiscovery]
Xu, X., Talaulikar, K., Bi, K., Tantsura, J., and N. Triantafillis, "BGP Neighbor Discovery", draft-xu-idr-neighbor-autodiscovery-12 (work in progress), November 2019.

[RFC0826] Plummer, D., "An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, DOI 10.17487/RFC0826, November 1982, <https://www.rfc-editor.org/info/rfc826>.

[RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, DOI 10.17487/RFC2385, August 1998, <https://www.rfc-editor.org/info/rfc2385>.

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

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., Ed., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, DOI 10.17487/RFC5082, October 2007, <https://www.rfc-editor.org/info/rfc5082>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <https://www.rfc-editor.org/info/rfc5925>.

[RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <https://www.rfc-editor.org/info/rfc8177>.

Appendix A. Analysis of Candidate Approaches

As part of the work on distilling the requirements for BGP auto-discovery, the Design Team reviewed several proposals for implementing auto-discovery. The analysis of these proposals, including missing elements the Design Team decided were part of the requirements, follows.

A.1. BGP Peer Discovery at Layer Two

BGP Discovery at Layer-2 would entail finding potential peers on a LAN or on Point-to-Point links and discovering their Layer-3 attributes, such as, IP addresses, etc.

There are two available candidates for peer discovery at Layer-2, one is based on Link Layer Discovery Protocol (LLDP) and the other is based on Layer 3 Discovery Protocol, L3DL [I-D.ietf-lsvr-l3dl].

A.1.1. LLDP based Approach

LLDP is a widely deployed protocol with implementations in most devices in data centers. Currently it only advertises the managment Layer-3 address, but could presumably be extended to include the per-interface addresses.

LLDP has a limitation that all information must fit in a single PDU (it does not support fragmentation / a "session"). There is an early LLDPv2 development effort to extend this in the IEEE.

[I-D.acee-idr-lldp-peer-discovery] describes how to use the LLDP IETF Organizationally Specific TLV to augment the LLDP TLV set to exchange BGP Config Sub-TLVs signaling:

- AFI
- IP address (IPv4 or IPv6)
- Local AS number
- Local BGP Identifier (AKA, BGP Router ID)
- Session Group-ID; i.e., the BGP Device Role
- BGP Session Capabilities

- Key Chain
- Local Address (as BGP Next Hop).

A.1.2. L3DL based Approach

L3DL [I-D.ietf-lsvr-l3dl] is an ongoing development in the IETF LSVR Working Group with the goal of discovering IP Layer-3 attributes of links, such as neighbor IP addressing, logical link IP encapsulation abilities, and link liveness which may then be disseminated for the use of BGP-SPF and similar protocols.

L3DL Upper Layer Protocol Configuration, [I-D.ietf-lsvr-l3dl-ulpc], details signaling the minimal set of parameters needed to start a BGP session with a discovered peer. Details such as loopback peering are handled by attributes in the L3DL protocol itself. The information which can be discovered by L3DL is:

- AS number
- Local IP address, IPv4 or IPv6, and
- BGP Authentication.

L3DL and L3DL-ULPC have well-specified security mechanisms, see [I-D.ietf-lsvr-l3dl-signing].

The functionality of L3DL-ULPC is similar but not quite the same as the needs of IDR Design Team. For example, L3DL is designed to meet more complex needs. L3DL's predecessor, LSOE [I-D.ietf-lsvr-lsoe], was simpler and might be a better candidate for adaptation. If needed, the design of LSOE may be customized for the needs of BGP peer auto- disovery.

Unlike LLDP, L3DL has only one implementation, and LSOE has only one open source implementation, and neither is significantly deployed.

A.2. Link-Local Discovery

Some existing BGP auto-configuration mechanisms leverage "point to point" addressing schemes to bootstrap BGP sessions. One example utilizes an IP subnet numbered such that it may contain only two hosts - for IPv4, a /30 or /31 network; for IPv6 a /127 network. An additional mechanism may leverage IPv4 ARP [RFC0826] or IPv6 Neighbor Discovery [RFC4861] to learn of hosts on a subnet.

Such existing mechanisms do not provide an auto-discovery protocol with necessary parameters. Rather, they simplify configuration by permitting BGP session configuration templates to be easily applied to interfaces without requiring addressing to be known a priori.

A.3. BGP peer Discovery at Layer Three

Discovery at Layer-3 can assume IP addressability, though the IP addresses of potential peers are not known a priori and need to be discovered before further negotiation. IP multicast may be a good choice to address the above concern.

The possible problem would appear to discovery at Layer-3 is that one may not know whether to use IPv4 or IPv6. This might be exacerbated by the possibility of a potential peer not being on the local subnet, and hence broadcast and similar techniques may not be applicable. While in data center network or networks in a single administrative domain, such issue could be easily solved.

If one can assume that the BGP session is based on point-to-point link, then discovery might try IPv6 link-local or even IPv4 link-local. A link broadcast or multicast protocol may also be used. For switched or bridged multi-point which is at least on the same subnet, VLAN, etc., multicast or broadcasts might be a viable approach.

There are four available candidates for BGP peer discovery at Layer-3: One is based on extending BGP with new Hello message for peer auto-discovery. One is based on reusing BGP OPEN message format for peer auto-discovery. One is based on bootstrapping BGP sessions via existing BGP sessions. One is based upon bootstraping a BGP Route Reflector via the OSPF protocol.

A.3.1. New BGP Hello Message based Approach

[I-D.xu-idr-neighbor-autodiscovery] describes a BGP neighbor discovery mechanism which is based on a newly defined UDP based BGP Hello message. The BGP Hello message is sent in multicast to discover the directly connected BGP peers. According to the message header format and the TLVs carried in the message, the information which can be signaled is:

- AS number
- BGP Identifier
- Accepted ASN list
- Peering address (IPv4 or IPv6)
- Local prefix (for loopback)
- Link attributes
- Neighbor state
- BGP Authentication

The mechanisms in this draft do not currently handle fragmentation.

The mechanism in this draft is perhaps unique among the other proposals in requiring bi-directional state.

A.3.2. BGP OPEN Message based Approach

[I-D.raszuk-idr-bgp-auto-session-setup] describes a BGP neighbor discovery mechanism by reusing BGP OPEN message format with newly defined UDP port. The message is called BGP Session Explorer (BSE) packet and is sent in multicast. Since the message format is the same as BGP OPEN, the information which can be signaled is:

- AS number
- BGP Identifier
- Peering address

The mechanism is currently under-specified with respect to a number of similar properties described elsewhere. A general implication is that those properties - and others providing for extensibility of the auto-discovery mechanism - would need to be added to the BGP OPEN message and deal with the related impacts on the BGP session's finite-state machine.

BGP PDUs, including the OPEN message, may be up to 4k in size. Since this mechanism leverages Layer 3 multicast, a PDU fragmentation mechanism may need to be described.

A.3.3. Bootstrapping BGP via BGP

[I-D.raszuk-idr-bgp-auto-discovery] describes a new BGP address family. The NLRI carries a Group ID + BGP Identifier as the key. A new BGP Path Attribute carries information about the sessions:

- AS Number
- AFI/SAFI
- BGP Identifier
- Peer Transport Address
- Flags to declare a session for information only, to force a reset of a session on parameter changes, etc.

Since the BGP auto-discovery state is carried by BGP, it inherits the security implications of the underlying BGP session.

PDU size considerations are identical to those of a BGP UPDATE message.

Similarly, extensibility considerations would rely on either the new BGP Path Attribute, or one yet to be defined.

A.3.4. Bootstrapping BGP via OSPF

[I-D.acee-ospf-bgp-rr] describes a mechanism to learn BGP Route Reflectors via OSPFv2/OSPFv3 LSAs. Multiple types of scopes are defined for these LSAs to help constrain where they are advertised in an OSPF domain.

The BGP Route Reflector TLV contains:

- Local AS Number
- IPv4 or IPv6 Address of the Route Reflector
- A list of AFI/SAFIs supported by the Route Reflector

The BGP Route Reflector TLV may be advertised more than once, potentially to describe different IP transport endpoints.

This mechanism does not provide for security properties of the BGP session or transport properties such as BFD or GTSM.

Authors' Addresses

Randy Bush
Arrcus, Inc. & Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, WA 98110
US

Email: randy@psg.com

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

Email: jie.dong@huawei.com

Jeffrey Haas (editor)
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
US

Email: jhaas@juniper.net

Warren Kumari (editor)
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Email: warren@kumari.net

IDR
Internet-Draft
Intended status: Standards Track
Expires: August 13, 2021

F. Qin
China Mobile
H. Yuan
UnionPay
T. Zhou
G. Fioccola
Y. Wang
Huawei

February 9, 2021

BGP SR Policy Extensions to Enable IFIT
draft-ietf-idr-sr-policy-ifit-01

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Status of This Memo

This Internet-Draft will expire on August 13, 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 (Operations, Administration, and Maintenance) 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 (Path Computation Element Communication Protocol), [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), [I-D.ietf-mpls-rfc6374-sfl], [I-D.gandhi-mpls-rfc6374-sr] and [I-D.gandhi-mpls-ioam-sr] for Segment Routing over the MPLS data plane (SR-MPLS).

4. IFIT Attributes in SR Policy

As defined in [I-D.ietf-idr-segment-routing-te-policy], a new SAFI is defined (the SR Policy SAFI with codepoint 73) as well as a new NLRI. The NLRI contains the SR Policy candidate path and, according to [I-D.ietf-idr-segment-routing-te-policy], 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
                    ...
                ...
```

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
                SRv6 Binding SID
                Preference
                Priority
                Policy Name
                Policy Candidate Path 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.

Once the IFIT attributes are signalled, if a packet arrives at the headend and, based on the types of steering described in [I-D.ietf-spring-segment-routing-policy], it may get steered into an SR Policy where IFIT methods are applied. Therefore it will be managed consequently with the corresponding IOAM or Alternate Marking information according to the enabled IFIT methods.

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.

In case of empty IFIT Attributes Sub-TLV, i.e. no further IFIT sub-TLV and Length=0, IFIT methods will not be activated. If two conflicting IOAM sub-TLVs are present (e.g. Pre-allocated Trace Option and Incremental Trace Option) it means that they are not usable and none of the two methods will be activated. The same applies if there is more than one instance of the sub-TLV of the same type. Anyway the validation of the individual fields of the IFIT Attributes sub-TLVs are handled by the SRPM (SR Policy Module).

The process of stopping IFIT methods can be done by setting empty IFIT Attributes Sub-TLV, while, for modifying IFIT methods parameters, the IFIT Attributes Sub-TLVs can be updated accordingly. Additionally the backward compatibility is guaranteed, since an implementation that does not understand IFIT Attributes Sub-TLV can simply ignore it.

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 and ignored on receipt.

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 definition 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].

Flags: A 16-bit field. The definition is the same as described in section 3.2 of [I-D.ietf-ippm-ioam-direct-export].

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].

Rsvd: A 4-bit field reserved for further usage. It MUST be zero and ignored on receipt.

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].

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   |H|E| R |
    +---------------------------------------+---------------+-------+
```

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.

H: A flag indicating that the measurement is Hop-By-Hop.

E: A flag indicating that the measurement is end to end.

R: A 2-bit field reserved for further usage. It MUST be zero and ignored on receipt.

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],indeed A BGP Speaker MUST perform the syntactic validation of the SR Policy NLRI to determine if it is malformed, including the TLVs/sub-TLVs. In case of any error detected, either at the attribute or its TLV/sub-TLV level, the "treat-as-withdraw" strategy MUST be applied.

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].

The 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-02 (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-11 (work in progress), November 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-21 (work in progress), January 2021.

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 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-02 (work in progress), November 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-03 (work in progress), October 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-04 (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.

[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>.

[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>.

[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-05 (work in progress), January 2021.

[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-08 (work in progress), December 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: August 19, 2021

K. Vairavakkalai
N. Venkataraman
B. Rajagopalan
Juniper Networks, Inc.
G. Mishra
Verizon Communications Inc.
M. Khaddam
Cox Communications Inc.
X. Xu
Alibaba Inc.
R. Szarecki
Google.

February 15, 2021

BGP Classful Transport Planes
draft-kaliraj-idr-bgp-classful-transport-planes-07

Abstract

This document specifies a mechanism, referred to as "service mapping", to express association of overlay routes with underlay routes satisfying a certain SLA, using BGP. The document describes a framework for classifying underlay routes into transport classes, and mapping service routes to specific transport class.

The "Transport class" construct maps to a desired SLA, and can be used to realize the "Topology Slice" in 5G Network slicing architecture.

This document specifies BGP protocol procedures that enable dissimination of such service mapping information that may span multiple co-operating administrative domains. These domains may be administetered by the same provider or closely co-ordinating provider networks.

It makes it possible to advertise multiple tunnels to the same destination address, thus avoiding need of multiple loopbacks on the egress node.

A new BGP transport layer address family (SAFI 76) is defined for this purpose that uses RFC-4364 technology and follows RFC-8277 NLRI encoding. This new address family is called "BGP Classful Transport", aka BGP CT.

It carries transport prefixes across tunnel domain boundaries (e.g. in Inter-AS Option-C networks), parallel to BGP LU (SAFI 4) . It dissiminates "Transport class" information for the transport prefixes

across the participating domains, which is not possible with BGP LU. This makes the end-to-end network a "Transport Class" aware tunneled network.

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 August 19, 2021.

Copyright Notice

Table of Contents

1. Introduction

To facilitate service mapping, the tunnels in a network can be grouped by the purpose they serve into a "Transport Class". The tunnels could be created using any signaling protocol, such as LDP, RSVP, BGP LU or SPRING. The tunnels could also use native IP or IPv6, as long as they can carry MPLS payload. Tunnels may exist between different pair of end points. Multiple tunnels may exist between the same pair of end points.

Thus, a Transport Class consists of tunnels created by various protocols that satisfy the properties of the class. For example, a "Gold" transport class may consist of tunnels that traverse the shortest path with fast re-route protection, a "Silver" transport class may hold tunnels that traverse shortest paths without protection, a "To NbrAS Foo" transport class may hold tunnels that exit to neighboring AS Foo, and so on.

The extensions specified in this document can be used to create a BGP transport tunnel that potentially spans domains, while preserving its Transport Class. Examples of domain are Autonomous System (AS), or IGP area. Within each domain, there is a second level underlay tunnel used by BGP to cross the domain. The second level underlay tunnels could be hetrogeneous: Each domain may use a different type of tunnel (e.g. MPLS, IP, GRE), or use a differnet signaling protocol. A domain boundary is demarcated by a rewrite of BGP nexthop to ’self’ while re-advertising tunnel routes in BGP. Examples of domain boundary are inter-AS links and inter-region ABRs. The path uses MPLS label-switching when crossing domain boundary and uses the native intra-AS tunnel of the desired transport class when traversing within a domain.

Overlay routes carry sufficient indication of the Transport Classes they should be encapsulated over, in form of BGP community called the "Mapping community". Based on the mapping community, "route resolution" procedure on the ingress node selects from the corresponding Transport Class an appropriate tunnel whose destination matches (LPM) the nexthop of the overlay route. If the overlay route is carried in BGP, the protocol nexthop (or, PNH) is generally carried as an attribute of the route.

The PNH of the overlay route is also referred to as "service endpoint" (SEP). The service endpoint may exist in the same domain as the service ingress node or lie in a different domain, adjacent or non-adjacent. In the former case, reachability to the SEP is provided by an intra-domain tunneling protocol, and in the latter case, reachability to the SEP is via BGP transport families.

In this architecture, the intra-domain transport protocols (e.g. RSVP, SRTE) are also "Transport Class aware", and they publish ingress routes in Transport RIB associated with the Transport Class, at the tunnel ingress node. These routes are then redistributed into BGP CT to be advertised to adjacent domains. It is outside the scope of this document how exactly the transport protocols are made transport class aware, though configuration on the tunnel ingress node is a simple mechanism to achieve it.

This document describes mechanisms to:

Model a "Transport Class" as "Transport RIB" on a router, consisting of tunnel ingress routes of a certain class.

Enable service routes to resolve over an intended Transport Class by virtue of carrying the appropriate "Mapping community". Which results in using the corresponding Transport RIB for finding nexthop reachability.

Advertise tunnel ingress routes in a Transport RIB via BGP without any path hiding, using BGP VPN technology and Add-path. Such that overlay routes in the receiving domains can also resolve over tunnels of associated Transport Class.

Provide a way for co-operating domains to reconcile any differences in extended community namespaces, and interoperate between different transport signaling protocols in each domain.

In this document we focus mainly on MPLS as the intra-domain transport tunnel forwarding, but the mechanisms described here would work in similar manner for non-MPLS (e.g. IP, GRE, UDP) transport tunnel forwarding technologies too.

This document assumes MPLS forwarding when crossing domain boundaries, as that is the defacto standard in deployed networks today. But mechanisms specified in this document can also support different forwarding technologies (e.g. SRv6). A future document may describe such adaptations, it is out of scope of this document.

The document Seamless Segment Routing [Seamless-SR] describes various use cases and applications of procedures described in this document.

2. Terminology

LSP: Label Switched Path.

TE : Traffic Engineering.

SN : Service Node.

BN : Border Node.

TN : Transport Node, P-router.

BGP-VPN : VPNs built using RFC4364 mechanisms.

RT : Route-Target extended community.

RD : Route-Distinguisher.

PNH : Protocol-Nexthop address carried in a BGP Update message.

SEP : Service End point, the PNH of a Service route.

LPM : Longest Prefix Match.

Service Family : BGP address family used for advertising routes for "data traffic", as opposed to tunnels.

Transport Family : BGP address family used for advertising tunnels, which are in turn used by service routes for resolution.

Transport Tunnel : A tunnel over which a service may place traffic. These tunnels can be GRE, UDP, LDP, RSVP, or SR-TE.

Tunnel Domain : A domain of the network containing SN and BN, under a single administrative control that has a tunnel between SN and BN. An end-to-end tunnel spanning several adjacent tunnel domains can be created by "stitching" them together using labels.

Transport Class : A group of transport tunnels offering the same type of service.

Transport Class RT : A Route-Target extended community used to identify a specific Transport Class.

Transport RIB : At the SN and BN, a Transport Class has an associted Transport RIB that holds its tunnel routes.

Transport Plane : An end to end plane comprising of transport tunnels belonging to same transport class. Tunnels of same transport class are stitched together by BGP route readvertisements with nexthop-self, to span across domain boundaries using Label-Swap forwarding mechanism similar to Inter-AS option-b.

Mapping Community : BGP Community/Extended-community on a service route, that maps it to resolve over a Transport Class.

3. Transport Class

A Transport Class is defined as a set of transport tunnels that share certain characteristics useful for underlay selection.

On the wire, a transport class is represented as the Transport Class RT, which is a new Route-Target extended community.

A Transport Class is configured at SN and BN, along with attributes like RD and Route-Target. Creation of a Transport Class instantiates the associated Transport RIB and a Transport routing instance to contain them all.

The operator may configure a SN/BN to classify a tunnel into an appropriate Transport Class, which causes the tunnel's ingress routes to be installed in the corresponding Transport RIB. At a BN, these tunnel routes may then be advertised into BGP CT.

Alternatively, a router receiving the transport routes in BGP with appropriate signaling information can associate those ingress routes to the appropriate Transport Class. E.g. for Classful Transport family (SAFI 76) routes, the Transport Class RT indicates the Transport Class. For BGP LU family(SAFI 4) routes, import processing based on Communities or inter-AS source-peer may be used to place the route in the desired Transport Class.

When the ingress route is received via SRTE [SRTE], which encodes the Transport Class as an integer 'Color' in the NLRI as "Color:Endpoint", the 'Color' is mapped to a Transport Class during import processing. SRTE ingress route for 'Endpoint' is installed in that transport class. The SRTE route when advertised out to BGP speakers will then be advertised in Classful Transport family with Transport Class RT and a new label. The MPLS swap route thus installed for the new label will pop the label and deliver decapsulated traffic into the path determined by SRTE route.

4. "Transport Class" Route Target Extended Community

This document defines a new type of Route Target, called "Transport Class" Route Target Extended Community.

"Transport Class" Route Target extended community is a transitive extended community EXT-COMM [RFC4360] of extended-type, with a new Format (Type high = 0xa) and SubType as 0x2 (Route Target).

This new Route Target Format has the following encoding:

```
    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= 0xa   | SubType= 0x02 |            Reserved           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Transport Class ID                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

"Transport Class" Route Target Extended Community

Type: 2 octets

Type field contains value 0xa.

SubType: 2 octets

Subtype field contain 0x2. This indicates 'Route Target'.

Transport Class ID: 4 octets

The least significant 32-bits of the value field contain the "Transport Class" identifier, which is a 32-bit integer.

The remaining 2 octets after SubType field are Reserved, they MUST be set to zero by originator, and ignored, left unaltered by receiver.

The "Transport class" Route Target Extended community follows the mechanisms for VPN route import, export as specified in BGP-VPN [RFC4364], and Route Target Contrain mechanisms as specified in VPN-RTC [RFC4684]

A BGP speaker that implements RT Constraint VPN-RTC [RFC4684] MUST apply the RT Constraint procedures to the "Transport class" Route Target Extended community as-well.

The Transport Class Route Target Extended community is carried on Classful Transport family routes, and allows associating them with appropriate Transport RIBs at receiving BGP speakers.

Use of the Transport Class Route Target Extended community with a new Type code avoids conflicts with any VPN Route Target assignments already in use for service families.

5. Transport RIB

A Transport RIB is a routing-only RIB that is not installed in forwarding path. However, the routes in this RIB are used to resolve reachability of overlay routes' PNH. Transport RIB is created when the Transport Class it represents is configured.

Overlay routes that want to use a specific Transport Class confine the scope of nexthop resolution to the set of routes contained in the corresponding Transport RIB. This Transport RIB is the "Routing Table" referred in Section 9.1.2.1 RFC4271 [1]

Routes in a Transport RIB are exported out in 'Classful Transport' address family.

6. Transport Routing Instance

A BGP VPN routing instance that is a container for the Transport RIB. It imports, and exports routes in this RIB with Transport Class RT. Tunnel destination addresses in this routing instance's context come from the "provider namespace". This is different from user VRFs for e.g., which contain prefixes in "customer namespace"

The Transport Routing instance uses the RD and RT configured for the Transport Class.

7. Nexthop Resolution Scheme

An implementation may provide an option for the service route to resolve over less preferred Transport Classes, should the resolution over preferred, or "primary" Transport Class fail.

To accomplish this, the set of service routes may be associated with a user-configured "resolution scheme", which consists of the primary Transport Class, and optionally, an ordered list of fallback Transport Classes.

A community called as "Mapping Community" is configured for a "resolution scheme". A Mapping community maps to exactly one resolution scheme. A resolution scheme comprises of one primary transport class and optionally one or more fallback transport classes.

A BGP route is associated with a resolution scheme during import processing. The first community on the route that matches a mapping community of a locally configured resolution scheme is considered the effective mapping community for the route. The resolution scheme thus found is used when resolving the route's PNH. If a route contains more than one mapping community, it indicates that the route considers these multiple mapping communities as equivalent. So the first community that maps to a resolution scheme is chosen.

A transport route received in BGP Classful Transport family SHOULD use a resolution scheme that contains the primary Transport Class without any fallback to best effort tunnels. The primary Transport Class is identified by the Transport Class RT carried on the route. Thus Transport Class RT serves as the Mapping Community for Classful Transport routes.

A service route received in a BGP service family MAY map to a resolution scheme that contains the primary Transport Class identified by the mapping community on the route, and a fallback to best effort tunnels transport class. The primary Transport Class is identified by the Mapping community carried on the route. For e.g. the Extended Color community may serve as the Mapping Community for service routes. Color:0:<n> MAY map to a resolution scheme that has primary transport class <n>, and a fallback to best-effort transport class.

8. BGP Classful Transport Family NLRI

The Classful Transport family will use the existing AFI of IPv4 or IPv6, and a new SAFI 76 "Classful Transport" that will apply to both IPv4 and IPv6 AFIs.

The "Classful Transport" SAFI NLRI itself is encoded as specified in https://tools.ietf.org/html/rfc8277#section-2 [RFC8277].

When AFI is IPv4 the "Prefix" portion of Classful Transport family NLRI consists of an 8-byte RD followed by an IPv4 prefix. When AFI is IPv6 the "Prefix" consists of an 8-byte RD followed by an IPv6 prefix.

Attributes on a Classful Transport route include the Transport Class Route-Target extended community, which is used to leak the route into the right Transport RIBs on SNs and BNs in the network.

9. Comparison with other families using RFC-8277 encoding

SAFI 128 (Inet-VPN) is a RF8277 encoded family that carries service prefixes in the NLRI, where the prefixes come from the customer namespaces, and are contexualized into separate user virtual service RIBs called VRFs, using RFC4364 procedures.

SAFI 4 (BGP LU) is a RFC8277 encoded family that carries transport prefixes in the NLRI, where the prefixes come from the provider namespace.

SAFI 76 (Classful Transport) is a RFC8277 encoded family that carries transport prefixes in the NLRI, where the prefixes come from the provider namespace, but are contexualized into separate Transport RIBs, using RFC4364 procedures.

It is worth noting that SAFI 128 has been used to carry transport prefixes in "L3VPN Inter-AS Carrier's carrier" scenario, where BGP LU/LDP prefixes in CsC VRF are advertised in SAFI 128 towards the remote-end baby carrier.

In this document a new AFI/SAFI is used instead of reusing SAFI 128 to carry these transport routes, because it is operationally advantageous to segregate transport and service prefixes into separate address families, RIBs. E.g. It allows to safely enable "per-prefix" label allocation scheme for Classful Transport prefixes without affecting SAFI 128 service prefixes which may have huge scale. "per prefix" label allocation scheme keeps the routing churn local during topology changes.

A new family also facilitates having a different readvertisement path of the transport family routes in a network than the service route readvertisement path. viz. Service routes (Inet-VPN) are exchanged over an EBGP multihop session between Autonomous systems with nexthop unchanged; whereas Classful Transport routes are readvertised over EBGP single hop sessions with "nexthop-self" rewrite over inter-AS links.

The Classful Transport family is similar in vein to BGP LU, in that it carries transport prefixes. The only difference is, it also carries in Route Target an indication of which Transport Class the transport prefix belongs to, and uses RD to disambiguate multiple instances of the same transport prefix in a BGP Update.

10. Protocol Procedures

This section summarizes the procedures followed by various nodes speaking Classful Transport family

Preparing the network for deploying Classful Transport planes

Operator decides on the Transport Classes that exist in the network, and allocates a Route-Target to identify each Transport Class.

Operator configures Transport Classes on the SNs and BNs in the network with unique Route-Distinguishers and Route-Targets.

Implementations may provide automatic generation and assignment of RD, RT values for a transport routing instance; they MAY also provide a way to manually override the automatic mechanism, in order to deal with any conflicts that may arise with existing RD, RT values in the different network domains participating in a deployment.

Origination of Classful Transport route:

At the ingress node of the tunnel’s home domain, the tunneling protocols install routes in the Transport RIB associated with the Transport Class the tunnel belongs to.

The ingress node then advertises this tunnel destination into BGP as a Classful Transport family route with NLRI RD:TunnelEndpoint, attaching a ’Transport Class’ Route Target that identifies the Transport Class. This BGP CT route is advertised to EBGP peers and IBGP peers which are RR-clients. This route MUST NOT be advertised to the IBGP peers who are not RR-clients.

Alternatively, the egress node of the tunnel i.e. the tunnel endpoint can originate the same BGP Classful Transport route, with NLRI RD:TunnelEndpoint and PNH TunnelEndpoint, which will resolve over the tunnel route at the ingress node. When the tunnel is up, the Classful Transport BGP route will become usable and get re-advertised.

Unique RD SHOULD be used by the originator of a Classful Transport route to disambiguate the multiple BGP advertisements for a transport end point.

Ingress node receiving Classful Transport route

On receiving a BGP Classful Transport route with a PNH that is not directly connected, e.g. an IBGP-route, a mapping community on the route (the Transport Class RT) indicates which Transport Class this route maps to. The routes in the associated Transport RIB are used to resolve the received PNH. If there does not exist a route in the Transport RIB matching the PNH, the Classful Transport route is considered unusable, and MUST NOT be re-advertised further.

Border node readvertising Classful Transport route with nexthop self:

The BN allocates an MPLS label to advertise upstream in Classful Transport NLRI. The BN also installs an MPLS swap-route for that label that swaps the incoming label with a label received from the downstream BGP speaker, or pops the incoming label. And then pushes received traffic to the transport tunnel or direct interface that the Classful Transport route's PNH resolved over.

The label SHOULD be allocated with "per-prefix" label allocation semantics. The prefix used as key is formed by stripping RD from the BGP CT NLRI prefix. This helps in avoiding BGP CT route churn through out the CT network when a failure happens in a domain. The failure is not propagated further than the BN closest to the failure.

The value of advertised MPLS label is locally significant, and is dynamic by default. The BN may provide option to allocate a value from a statically carved out range. This can be achieved using locally configured export policy, or via mechanisms described in BGP Prefix-SID [RFC8669].

Border node receiving Classful Transport route on EBGP :

If the route is received with PNH that is known to be directly connected, e.g. EBGP single-hop peering address, the directly connected interface is checked for MPLS forwarding capability. No other nexthop resolution process is performed, as the inter-AS link can be used for any Transport Class.

If the inter-AS links should honor Transport Class, then the BN SHOULD follow procedures of an Ingress node described above, and perform nexthop resolution process. The interface routes SHOULD be installed in the Transport RIB belonging to the associated Transport Class.

Avoiding path-hiding through Route Reflectors

When multiple BNs exist that advertise a RDn:PEn prefix to RRs, the RRs may hide all but one of the BNs, unless ADDPATH [RFC7911] is used for the Classful Transport family. This is similar to L3VPN option-B scenarios. Hence ADDPATH SHOULD be used for Classful Transport family, to avoid path-hiding through RRs.

Avoiding loop between Route Reflectors in forwarding path

Pair of redundant ABRs acting as RR with nexthop-self may chose each other as best path instead of the upstream ASBR, causing a traffic forwarding loop.

Implementations SHOULD provide a way to alter the tie-breaking rule specified in BGP RR [RFC4456] to tie-break on CLUSTER_LIST step before ROUTER-ID step, when performing path selection for BGP CT routes. RFC4456 considers pure RR which is not in forwarding path. When RR is in forwarding path and reflects routes with nexthop-self, which is the case for ABR BNs in a BGP transport network, this rule may cause loops. This document suggests the following modification to the BGP Decision Process Tie Breaking rules (Sect. 9.1.2.2, [RFC4271]) when doing path selection for BGP CT family routes:

The following rule SHOULD be inserted between Steps e) and f): a BGP Speaker SHOULD prefer a route with the shorter CLUSTER_LIST length. The CLUSTER_LIST length is zero if a route does not carry the CLUSTER_LIST attribute.

Some deployment considerations can also help in avoiding this problem:

IGP metric should be assigned such that "ABR to redundant ABR" cost is inferior than "ABR to upstream ASBR" cost.

Tunnels belonging to special Transport classes SHOULD NOT be provisioned between ABR to ABRs. This will ensure that the route received from an ABR with nexthop-self will not be usable at a redundant ABR.

This avoids possibility of such loops altogether, irrespective of whether the path selection modification mentioned above is implemented.

Ingress node receiving service route with mapping community

Service routes received with mapping community resolve using Transport RIBs determined by the resolution scheme. If the resolution process does not find an usable Classful Transport route or tunnel route in any of the Transport RIBs, the service route MUST be considered unusable for forwarding purpose.

Coordinating between domains using different community namespaces.

Cooperating option-C domains may sometimes not agree on RT, RD, Mapping-community or Transport Route Target values because of differences in community namespaces; e.g. during network mergers or renumbering for expansion. Such deployments may deploy mechanisms to map and rewrite the Route-target values on domain boundaries, using per ASBR import policies. This is no different than any other BGP VPN family. Mechanisms employed in inter-AS

VPN deployments may be used with the Classful Transport family also.

The resolution schemes SHOULD allow association with multiple mapping communities. This helps with renumbering, network mergers, or transitions.

Though RD can also be rewritten on domain boundaries, deploying unique RDs is strongly RECOMMENDED, because it helps in trouble shooting by uniquely identifying originator of a route, and avoids path-hiding.

This document defines a new format of Route-Target extended-community to carry Transport Class, this avoids collision with regular Route Target namespace used by service routes.

11. Scaling considerations

11.1. Avoiding unintended spread of CT routes across domains.

RFC8212 [RFC8212] suggests BGP speakers require explicit configuration of both BGP Import and Export Policies for any EBGP sessions, in order to receive or send routes on EBGP sessions.

It is recommended to follow this for BGP CT routes. It will prohibit unintended advertisement of transport routes through out the BGP CT transport domain which may span multiple AS. This will conserve usage of MPLS label and nexthop resources in the network. An ASBR of a domain can be provisioned to allow routes with only the Transport targets that are required by SNs in the domain.

11.2. Constrained distribution of PNHs to SNs (On Demand Nexthop)

This section describes how the number of Protocol Nexthops advertised to a SN or BN can be constrained using BGP Classsful Transport and VPN RTC [RFC4684]

An egress SN MAY advertise BGP CT route for RD:eSN with two Route Targets: transport-target:0:<TC> and a RT carrying <eSN>:<TC>. Where TC is the Transport Class identifier, and eSN is the IP-address used by SN as BGP nexthop in it's service route advertisements.

transport-target:0:<TC> is the new type of route target (Transport Class RT) defined in this document. It is carried in BGP extended community attribute (BGP attribute code 16).

The RT carrying <eSN>:<TC> MAY be an IP-address specific regular RT (BGP attribute code 16), IPv6-address specific RT (BGP attribute code 25), or a Wide-communities based RT (BGP attribute code 34) as described in RTC-Ext [RTC-Ext]

An ingress SN MAY import BGP CT routes with Route Target carrying: <eSN>:<TC>. The ingress SN MAY learn the eSN values either by configuration, or it MAY discover them from the BGP nexthop field in the BGP VPN service routes received from eSN. A BGP ingress SN receiving a BGP service route with nexthop of eSN SHOULD generate a RTC/Extended-RTC route for Route Target prefix <Origin ASN>:<eSN>/[80|176] in order to learn BGP CT transport routes to reach eSN. This allows constrained distribution of the transport routes to the PNHs actually required by iSN.

When path of route propogation of BGP CT routes is same as the RTC routes, a BN would learn the RTC routes advertised by ingress SNs and propagate further. This will allow constraining distribution of BGP CT routes for a PNH to only the necessary BNs in the network, closer to the egress SN.

This mechanism provides "On Demand Nexthop" of BGP CT routes, which help with scaling of MPLS forwarding state at SN and BN.

But the amount of state carried in RTC family may become proportional to number of PNHs in the network. To strike a balance, the RTC route advertisements for <Origin ASN>:<eSN>/[80|176] MAY be confined to the BNs in home region of ingress-SN, or the BNs of a super core.

Such a BN in the core of the network SHOULD import BGP CT routes with Transport Class Route Target: 0:<TC>, and generate a RTC route for <Origin ASN>:0:<TC>/96, while not propagating the more specific RTC requests for specific PNHs. This will let the BN learn transport routes to all eSN nodes. But confine their propagation to ingress-SNs.

11.3. Limiting scope of visibility of PE loopback as PNHs

It may be even more desirable to limit the number of PNHs that are globaly visible in the network. This is possible using mechanism described in MPLS Namespaces [MPLS-NAMESPACES]

Such that advertisement of PE loopback addresses as next-hop in BGP service routes is confined to the region they belong to. An anycast IP-address called "Context Protocol Nexthop Address" abstracts the PEs in a region from other regions in the network, swapping the PE scoped service label with a CPNH scoped private namespace label.

This provides much greater advantage in terms of scaling and convergence. Changes to implement this feature are required only on the region’s BNs and RR.

12. OAM considerations

Standard MPLS OAM procedures specified in [RFC8029] also apply to BGP Classful Transport.

The ’Target FEC Stack’ sub-TLV for IPv4 Classful Transport has a Sub-Type of [TBD], and a length of 13. The Value field consists of the RD advertised with the Classful Transport prefix, the IPv4 prefix (with trailing 0 bits to make 32 bits in all), and a prefix length, 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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      Route Distinguisher                      |
    |                          (8 octets)                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         IPv4 prefix                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Prefix Length |                 Must Be Zero                  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Classful Transport IPv4 FEC

The ’Target FEC Stack’ sub-TLV for IPv6 Classful Transport has a Sub-Type of [TBD], and a length of 25. The Value field consists of the RD advertised with the Classful Transport prefix, the IPv6 prefix (with trailing 0 bits to make 128 bits in all), and a prefix length, 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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                     Route Distinguisher                       |
       |                         (8 octets)                            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                          IPv6 prefix                          |
       |                                                               |
       |                                                               |
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       | Prefix Length |                 Must Be Zero                  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Classful Transport IPv6 FEC

13. Applicability to Network Slicing

In Network Slicing, the Transport Slice Controller (TSC) sets up the Topology (e.g. RSVP, SR-TE tunnels with desired characteristics) and resources (e.g. polices/shapers) in a transport network to create a Transport slice. The Transport class construct described in this document represents the "Topology Slice" portion of this equation.

The TSC can use the Transport Class Identifier (Color value) to provision a transport tunnel in a specific Topology Slice.

Further, Network slice controller can use the Mapping community on the service route to map traffic to the desired Transport slice.

14. Illustration of procedures with example topology

14.1. Topology

```
                   [RR26]      [RR27]                       [RR16]
                    |            |                             |
                    |            |                             |
                    |+-[ABR23]--+|+--[ASBR21]---[ASBR13]-+|+--[PE11]--+
                    ||          |||          `  /        |||          |
   [CE41]--[PE25]--[P28]       [P29]          `/        [P15]     [CE31]
                    |           | |           /`         | |          |
                    |           | |          /  `        | |          |
                    |           | |         /    `       | |          |
                    +--[ABR24]--+ +--[ASBR22]---[ASBR14]-+ +--[PE12]--+


          |                |                  |                    |
          +                +                  +                    +
       CE |     region-1   |  region-2        |                    |CE
      AS4              ...AS2...                       AS1          AS3

   41.41.41.41  ------------ Traffic Direction ---------->   31.31.31.31
```

This example shows a provider network that comprises of two Autonomous systems, AS1, AS2. They are serving customers AS3, AS4 respectively. Traffic direction being described is CE41 to CE31. CE31 may request a specific SLA, e.g. Gold for this traffic, when traversing these provider networks.

AS2 is further divided into two regions. So there are three tunnel domains in provider space. AS1 uses ISIS Flex-Algo intra-domain tunnels, whereas AS2 uses RSVP intra-domain tunnels.

The network has two Transport classes: Gold with transport class id 100, Bronze with transport class id 200. These transport classes are provisioned at the PEs and the Border nodes (ABRs, ASBRs) in the network.

Following tunnels exist for Gold transport class.

PE25_to_ABR23_gold - RSVP tunnel

PE25_to_ABR24_gold - RSVP tunnel

ABR23_to_ASBR22_gold - RSVP tunnel

ASBR13_to_PE11_gold - ISIS FlexAlgo tunnel

ASBR14_to_PE11_gold - ISIS FlexAlgo tunnel

Following tunnels exist for Bronze transport class.

PE25_to_ABR23_bronze - RSVP tunnel

ABR23_to_ASBR21_bronze - RSVP tunnel

ABR23_to_ASBR22_bronze - RSVP tunnel

ABR24_to_ASBR21_bronze - RSVP tunnel

ASBR13_to_PE12_bronze - ISIS FlexAlgo tunnel

ASBR14_to_PE11_bronze - ISIS FlexAlgo tunnel

These tunnels are either provisioned or auto-discovered to belong to transport class 100 or 200.

14.2. Service Layer route exchange

Service nodes PE11, PE12 negotiate service families (SAFI 1, 128) on the BGP session with RR16. Service helpers RR16, RR26 have multihop EBGP session to exchange service routes between the two AS. Similarly PE25 negotiates service families with RR26.

Forwarding happens using service routes at service nodes PE25, PE11, PE12 only. Routes received from CEs are not present in any other nodes' FIB in the network.

CE31 advertises a route for example prefix 31.31.31.31 with nexthop self to PE11, PE12. CE31 can attach a mapping community Color:0:100 on this route, to indicate its request for Gold SLA. Or, PE11 can attach the same using locally configured policies. Let us assume CE31 is getting VPN service from PE25.

The 31.31.31.31 route is readvertised in SAFI 128 by PE11 with nexthop self (1.1.1.1) and label V-L1, to RR16 with the mapping community Color:0:100 attached. This SAFI 128 route reaches PE25 via RR16, RR26 with the nexthop unchanged, as PE11 and label V-L1. Now PE25 can resolve the PNH 1.1.1.1 using transport routes received in BGP CT or BGP LU.

The IP FIB at PE25 will have a route for 31.31.31.31 with a nexthop thus found, that points to a Gold tunnel in ingress domain.

14.3. Transport Layer route propagation

ASBR13 negotiates BGP CT family with transport ASBRs ASBR21, ASBR22. They negotiate BGP CT family with RR27 in region 2. ABR23, ABR24 negotiate BGP CT family with RR27 in region 2 and RR26 in region 1. PE25 receives BGP CT routes from RR26. BGP LU family is also negotiated on these sessions alongside BGP CT family. BGP LU carries "best effort" transport class routes, BGP CT carries gold, bronze transport class routes.

ASBR13 is provisioned with transport class 100, RD value 1.1.1.3:10 and a transport route target 0:100. And a Transport class 200 with RD value 1.1.1.3:20, and transport route target 0:200.

Similarly, these transoprt classes are also configured on ASBRs, ABRs and PEs, with same transport route target, but unique RDs.

Ingress route for ASBR13_to_PE11_gold is advertised by ASBR13 in BGP CT family to ASBRs ASBR21, ASBR22. This route is sent with a NLRI containing RD prefix 1.1.1.3:10:1.1.1.1, Label B-L1 and a route target extended community transport-target:0:100. MPLS swap route is installed at ASBR13 for B-L1 with a nexthop pointing to ASBR13_to_PE11_gold tunnel.

Ingress route for ASBR13_to_PE11_bronze is advertised by ASBR13 in BGP CT family to ASBRs ASBR21, ASBR22. This route is sent with a NLRI containing RD prefix 1.1.1.3:20:1.1.1.1, Label B-L2 and a route target extended community transport-target:0:200. MPLS swap route is installed at ASBR13 for label B-L2 with a nexthop pointing to ASBR13_to_PE11_bronze tunnel

ASBR21 receives BGP CT route 1.1.1.3:10:1.1.1.1 over the single hop EBGP sesion, and readvertises with nexthop self (loopback adderss 2.2.2.1) to RR27, advertising a new label B-L3. MPLS swap route is installed for label B-L3 at ASBR21 to swap to received label B-L1 and forwards to ASBR13. RR27 readvertises this BGP CT route to ABR23, ABR24.

ASBR22 receives BGP CT route 1.1.1.3:10:1.1.1.1 over the single hop EBGP sesion, and readvertises with nexthop self (loopback adderss 2.2.2.2) to RR27, advertising a new label B-L4. MPLS swap route is installed for label B-L4 at ASBR21 to swap to received label B-L2 and forwards to ASBR13. RR27 readvertises this BGP CT route to ABR23, ABR24.

Addpath is enabled for BGP CT family on the sessions between RR27 and ASBRs, ABRs. Such that routes for 1.1.1.3:10:1.1.1.1 with the nexthops ASBR21 and ASBR22 are reflected to ABR23, ABR24 without any path hiding. Thus giving ABR23 visibiity of both available nexthops for Gold SLA.

ABR23 receives the route with nexthop 2.2.2.1, label B-L3 from RR27. The route target "transport-target:0:100" on this route acts as mapping community, and instructs ABR23 to strictly resolve the nexthop using transport class 100 routes only. ABR23 is unable to find a route for 2.2.2.1 with transport class 100. Thus it considers this route unusable and does not propagate it further. This prunes ASBR21 from Gold SLA tunneled path.

ABR23 also receives the route with nexthop 2.2.2.2, label B-L4 from RR27. The route target "transport-target:0:100" on this route acts as mapping community, and instructs ABR23 to strictly resolve the nexthop using transport class 100 routes only. ABR23 successfully resolves the nexthop to point to ABR23_to_ASBR22_gold tunnel. ABR23 readvertises this route with nexthop self (loopback address 2.2.2.3) and a new label B-L5 to RR26. Swap route for B-L5 is installed by ABR23 to swap to label B-L4, and forward into ABR23_to_ASBR22_gold tunnel.

RR26 reflects the route from ABR23 to PE25. PE25 receives the BGP CT route for prefix 1.1.1.3:10:1.1.1.1 with label B-L5, nexthop 2.2.2.3 and transport-target:0:100 from RR26. And it similarly resolves the nexthop 2.2.2.3 over transport class 100, pushing labels associated with PE25_to_ABR23_gold tunnel.

In this manner, the Gold transport LSP "ASBR13_to_PE11_gold" in egress-domain is extended by BGP CT until the ingress-node PE25 in ingress domain, to create an end-to-end Gold SLA path. MPLS swap routes are installed at ASBR13, ASBR22 and ABR23, when propagating the PE11 BGP CT Gold transport class route 1.1.1.3:10:1.1.1.1 with nexthop self towards PE25.

The BGP CT LSP thus formed, originates in PE25, and terminates in ASBR13, traversing over the Gold underlay LSPs in each domain. ASBR13 uses UHP to stitch the BGP CT LSP into the "ASBR13_to_PE11_gold" LSP to traverse the last domain, thus satisfying Gold SLA end-to-end.

When PE25 receives service route with nexthop 1.1.1.1 and mapping community Color:0:100, it resolves over this BGP CT route 1.1.1.3:10:1.1.1.1. Thus pushing label B-L5, and pushing as top label the labels associated with PE25_to_ABR23_gold tunnel.

14.4. Data plane view

14.4.1. Steady state

This section describes how the data plane looks like in steady state.

CE41 transmits an IP packet with destination as 31.31.31.31. On receiving this packet PE25 performs a lookup in the IP FIB associated with the CE41 interface. This lookup yeids the service route that pushes the VPN service label V-L1, BGP CT label B-L5, and labels for PE25_to_ABR23_gold tunnel. Thus PE25 encapsulates the IP packet in MPLS packet with label V-L1(innermost), B-L5, and top label as PE25_to_ABR23_gold tunnel. This MPLS packet is thus transmitted to ABR23 using Gold SLA.

ABR23 decapsulates the packet received on PE25_to_ABR23_gold tunnel as required, and finds the MPLS packet with label B-L5. It performs lookup for label B-L5 in the global MPLS FIB. This yields the route that swaps label B-L5 with label B-L4, and pushes top label provided by ABR23_to_ASBR22_gold tunnel. Thus ABR23 transmits the MPLS packet with label B-L4 to ASBR22, on a tunnel that satisfies Gold SLA.

ASBR22 similarly performs a lookup for label B-L4 in global MPLS FIB, finds the route that swaps label B-L4 with label B-L2, and forwards to ASBR13 over the directly connected MPLS enabled interface. This interface is a common resource not dedicated to any specific transport class, in this example.

ASBR13 receives the MPLS packet with label B-L2, and performs a lookup in MPLS FIB, finds the route that pops label B-L2, and pushes labels associated with ASBR13_to_PE11_gold tunnel. This transmits the MPLS packet with VPN label V-L1 to PE11, using a tunnel that preserves Gold SLA in AS 1.

PE11 receives the MPLS packet with V-L1, and performs VPN forwarding. Thus transmitting the original IP payload from CE41 to CE31. The payload has traversed path satisfying Gold SLA end-to-end.

14.4.2. Absorbing failure of primary path

This section describes how the data plane reacts when gold path experiences a failure.

Let us assume tunnel ABR23_to_ASBR22_gold goes down, such that now end-to-end Gold path does not exist in the network. This makes the BGP CT route for RD prefix 1.1.1.1:10:1.1.1.1 unusable at ABR23. This makes ABR23 send a BGP withdrawal for 1.1.1.1:10:1.1.1.1 to RR26, which then withdraws the prefix from PE25.

Withdrawal for 1.1.1.1:10:1.1.1.1 allows PE25 to react to the loss of gold path to 1.1.1.1. Let us assume PE25 is provisioned to use best-effort transport class as the backup path. This withdrawal of BGP CT route allows PE25 to adjust the nexthop of the VPN Service-route to push the labels provided by the BGP LU route. That repairs the traffic to go via best effort path. PE25 can also be provisioned to use Bronze transport class as the backup path. The repair will happen in similar manner in that case as-well.

Traffic repair to absorb the failure happens at ingress node PE25, in a service prefix scale independent manner. This is called PIC (Prefix scale Independent Convergence). The repair time will be proportional to time taken for withdrawing the BGP CT route.

15. IANA Considerations

This document makes following requests of IANA.

15.1. New BGP SAFI

New BGP SAFI code for "Classful Transport". Value 76.

This will be used to create new AFI,SAFI pairs for IPv4, IPv6 Classful Transport families. viz:

- "Inet, Classful Transport". AFI/SAFI = "1/76" for carrying IPv4 Classful Transport prefixes.
- "Inet6, Classful Transport". AFI/SAFI = "2/76" for carrying IPv6 Classful Transport prefixes.

15.2. New Format for BGP Extended Community

Please assign a new Format (Type high = 0xa) of extended community EXT-COMM [RFC4360] called "Transport Class" from the following registries:

the "BGP Transitive Extended Community Types" registry, and

the "BGP Non-Transitive Extended Community Types" registry.

Please assign the same low-order six bits for both allocations.

This document uses this new Format with subtype 0x2 (route target), as a transitive extended community.

The Route Target thus formed is called "Transport Class" route target extended community.

Taking reference of RFC7153 [RFC7153] , following requests are made:

15.2.1. Existing registries to be modified

15.2.1.1. Registries for the "Type" Field

15.2.1.1.1. Transitive Types

This registry contains values of the high-order octet (the "Type" field) of a Transitive Extended Community.

Registry Name: BGP Transitive Extended Community Types

```
      TYPE VALUE       NAME
+      0x0a             Transitive Transport Class Extended
+                       Community (Sub-Types are defined in the
+                       "Transitive Transport Class Extended
+                       Community Sub-Types" registry)
```

15.2.1.1.2. Non-Transitive Types

This registry contains values of the high-order octet (the "Type" field) of a Non-transitive Extended Community.

Registry Name: BGP Non-Transitive Extended Community Types

```
     TYPE VALUE       NAME

+     0x4a             Non-Transitive Transport Class Extended
+                      Community (Sub-Types are defined in the
+                      "Non-Transitive Transport Class Extended
+                      Community Sub-Types" registry)
```

15.2.2. New registries to be created

15.2.2.1. Transitive "Transport Class" Extended Community Sub-Types Registry

This registry contains values of the second octet (the "Sub-Type" field) of an extended community when the value of the first octet (the "Type" field) is 0x07.

Registry Name: Transitive Transport Class Extended
Community Sub-Types

RANGE	REGISTRATION PROCEDURE
0x00-0xBF	First Come First Served
0xC0-0xFF	IETF Review

SUB-TYPE VALUE	NAME
0x02	Route Target

15.2.2.2. Non-Transitive "Transport Class" Extended Community Sub-Types Registry

This registry contains values of the second octet (the "Sub-Type" field) of an extended community when the value of the first octet (the "Type" field) is 0x47.

Registry Name: Non-Transitive Transport Class Extended
Community Sub-Types

RANGE	REGISTRATION PROCEDURE
0x00-0xBF	First Come First Served
0xC0-0xFF	IETF Review

SUB-TYPE VALUE	NAME
0x02	Route Target

15.3. MPLS OAM code points

The following two code points are sought for Target FEC Stack sub-TLVs:

- IPv4 BGP Classful Transport

- IPv6 BGP Classful Transport

16. Security Considerations

Mechanisms described in this document carry Transport routes in a new BGP address family. That minimizes possibility of these routes leaking outside the expected domain or mixing with service routes.

When redistributing between SAFI 4 and SAFI 76 Classful Transport routes, there is a possibility of SAFI 4 routes mixing with SAFI 1 service routes. To avoid such scenarios, it is RECOMMENDED that implementations support keeping SAFI 4 routes in a separate transport RIB, distinct from service RIB that contain SAFI 1 service routes.

17. Acknowledgements

The authors thank Jeff Haas, John Scudder, Navaneetha Krishnan, Ravi M R, Chandrasekar Ramachandran, Shradha Hegde, Richard Roberts, Krzysztof Szarkowicz, John E Drake, Srihari Sangli, Vijay Kestur, Santosh Kolenchery, Robert Raszuk, Ahmed Darwish for the valuable discussions and review comments.

The decision to not reuse SAFI 128 and create a new address-family to carry these transport-routes was based on suggestion made by Richard Roberts and Krzysztof Szarkowicz.

18. References

18.1. Normative References

[MPLS-NAMESPACES]
Vairavakkalai, Ed., "Private MPLS-label namespaces", 08 2020, <https://tools.ietf.org/html/draft-kaliraj-bess-bgp-sig-private-mpls-labels-01#section-6.1>.

[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>.

[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>.

[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>.

[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>.

[RFC7153] Rosen, E. and Y. Rekhter, "IANA Registries for BGP Extended Communities", RFC 7153, DOI 10.17487/RFC7153, March 2014, <https://www.rfc-editor.org/info/rfc7153>.

[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>.

[RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <https://www.rfc-editor.org/info/rfc8029>.

[RFC8212] Mauch, J., Snijders, J., and G. Hankins, "Default External BGP (EBGP) Route Propagation Behavior without Policies", RFC 8212, DOI 10.17487/RFC8212, July 2017, <https://www.rfc-editor.org/info/rfc8212>.

[RFC8277] Rosen, E., "Using BGP to Bind MPLS Labels to Address Prefixes", RFC 8277, DOI 10.17487/RFC8277, October 2017, <https://www.rfc-editor.org/info/rfc8277>.

[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>.

[RTC-Ext] Zhang, Z., Ed., "Route Target Constrain Extension", 07 2020, <https://tools.ietf.org/html/draft-zzhang-idr-bgp-rt-constrains-extension-00#section-2>.

[Seamless-SR] Hegde, Ed., "Seamless Segment Routing", 11 2020, <https://datatracker.ietf.org/doc/html/draft-hegde-spring-mpls-seamless-sr-03>.

[SRTE] Previdi, S., Ed., "Advertising Segment Routing Policies in BGP", 11 2019, <https://tools.ietf.org/html/draft-ietf-idr-segment-routing-te-policy-08>.

18.2. URIs

[1] https://www.rfc-editor.org/rfc/rfc4271#section-9.1.2.1

Authors' Addresses

Kaliraj Vairavakkalai
Juniper Networks, Inc.
1133 Innovation Way,
Sunnyvale, CA 94089
US

Email: kaliraj@juniper.net

Natrajan Venkataraman
Juniper Networks, Inc.
1133 Innovation Way,
Sunnyvale, CA 94089
US

Email: natv@juniper.net

Balaji Rajagopalan
Juniper Networks, Inc.
Electra, Exora Business Park˜Marathahalli - Sarjapur Outer
Ring Road,
Bangalore, KA 560103
India

Email: balajir@juniper.net

Gyan Mishra
Verizon Communications Inc.
13101 Columbia Pike
Silver Spring, MD 20904
USA

Email: gyan.s.mishra@verizon.com

Mazen Khaddam
Cox Communications Inc.
Atlanta, GA
USA

Email: mazen.khaddam@cox.com

Xiaohu Xu
Alibaba Inc.
Beijing
China

Email: xiaohu.xxh@alibaba-inc.com

Rafal Jan Szarecki
Google.
1160 N Mathilda Ave, Bldg 5,
Sunnyvale,, CA 94089
USA

Email: szarecki@google.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: September 8, 2021

A. Khare, Ed.
Ciena Corporation
P. Bergeon, Ed.
Nokia
V. Kestur
Juniper Networks, Inc.
L. Jalil
Verizon
K. Kasavchenko

March 7, 2021

BGP FlowSpec Payload Matching
draft-khare-idr-bgp-flowspec-payload-match-08

Abstract

The rise in frequency, volume, and pernicious effects of DDoS attacks has elevated them from fare for the specialist to generalist press. Numerous reports detail the taxonomy of DDoS attacks, the varying motivations of their attackers, as well as the resulting impact for their targets ranging from internet or business services to network infrastrutures.

BGP FlowSpec (RFC 5575, "Dissemination of Flow Specification Rules") can be used to rapidly disseminate filtering rules to mitigate (distributed) denial-of-service (DoS) attacks. Operators can use existing FlowSpec components to match typical n-tuple criteria in pre-defined packet header fields such as IP protocol, IP prefix or port number. Recent enhancements to IP Router forwarding plane filter implementations also allow matches at arbitrary locations within the packet header or payload. This capability can be used to essentially match a signature for the attack traffic and can be combined with traditional n-tuple filter criteria to mitigate volumetric DDoS attacks and reduce false positive to a minimum.

To support this new filtering capability we define a new FlowSpec component, "Flexible Match Conditions", with similar matching semantics to those of existing components. This component will allow the operator to define a new match condition using a combination of offset and pattern values.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

BGP FlowSpec [RFC5575] can be used to rapidly disseminate filtering rules to mitigate (distributed) denial-of-service (DoS) attacks. Operators can use existing FlowSpec components to match typical n-tuple criteria in pre-defined packet header fields such as IP protocol, IP prefix and port number.

Recent enhancements to IP Router forwarding plane filter implementations also allow matches at arbitrary locations within the packet header or payload. This capability can be used to essentially match a signature for the attack traffic and can be combined with traditional n-tuple filter criteria to mitigate volumetric DDoS attacks and reduce false positive to a minimum.

To support this new filtering capability we define a new FlowSpec component, "Flexible Match Conditions", with similar matching semantics to those of existing components. This component will allow the operator to define a new match condition using a combination of offset and pattern values.

2. Definitions of Terms Used in This Memo

AFI - Address Family Identifier.

SAFI - Subsequent Address Family Identifier.

NLRI - Network Layer Reachability Information.

Flow specification controller - BGP speaker sending the flow specification rules to the IP edge routers (e.g. DDoS controllers).

Maximum Readable Length - The packet length in bits that a forwarding implementation can parse and make available for filtering. Abbreviated as MRL.

Maximum Pattern Length - The pattern length in bits that a forwarding implementation can match against the packet header or payload. Abbreviated as MPL.

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

3. Motivation

BGP FlowSpec couples both the advertisement of NLRI-specific match conditions, as well as the forwarding instance to which the filter is attached. This makes sense since BGP FlowSpec advertisements are most commonly generated, or at least verified, by human operators. The operator finds it intuitive to configure match conditions as human-readable values, native to each address family.

It is much friendlier, for instance, to define a filter that matches a source address of 192.168.1.1/32, than it is to work with the equivalent binary representation of that IPv4 address. Further, it is easier to use field names such as 'IPv4 source address' as part of the match condition, than it is to demarc that field using byte and bit offsets.

However, there are a number of use cases that benefit from the latter, more machine-readable approach.

3.1. Machine analysis of DDoS attacks

Volumetric DDoS attacks can severely impact services and network operator infrastructures but are also easily mitigated once identified. The challenge lies in fishing out a generally unvarying attack signature from a data stream. Machine analysis can be particularly useful here given the size of input involved in order to identify a pattern within the attack traffic flows.

Below we illustrate the need for the suggested approach with two use cases.

3.1.1. Matching based on payload

Volumetric DDoS attacks can either directly send traffic to a target or use reflection/amplification protocols to overload that target.

Reflection/amplification attacks are often identified by the UDP source port of a service that reflects and amplifies the attack traffic. However, blocking traffic based on source port can lead to further service interruption and eventually complete the attack especially in case of essential protocols such as NTP. There alo exist DDoS attack methodologies such as SSDP Diffraction or BitTorent amplification where values in most of layer 3 and layer 4 header fields, including source and destination UDP ports, are varied. That makes it challenging to mitigate based on existing Flow Specification components. At the same time these attacks often have a constant pattern in payload that can be used as matching criteria to further mitigate such DDoS attacks.

Direct attacks may also use a constant pattern in payload which can be used as a match criteria in filtering rules.

3.1.2. Matching based on any protocol header field or across fields

BGP FlowSpec [RFC5575] defines 12 Flow Specification component types that can be used to match traffic. However, a DDoS attack might result in illegitimate traffic with a specific pattern in a layer 3 or layer 4 header, and this pattern may not have a respective FlowSpec component type defined. The flexible match patterns defined in this document avoid extending BGP FlowSpec [RFC5575] with all theoretically possible header fields and allow matching across fields for any bitmask combinations.

3.2. Tunneled traffic

Tunnels continue to proliferate due to the benefits they provide. They can help reduce state in the underlay network. Tunnels allow bypassing routing decisions of the transit network. Traffic that is tunneled is often done so to obscure or secure. Common tunnel types include IPsec [RFC4301], Generic Routing Encapsulation (GRE) [RFC2890], et al.

By definition, transit nodes that are not the endpoints of the tunnel hold no attendant control or management plane state. These very qualities make it challenging to filter tunneled traffic at non-endpoints and it is usually infeasible to filter based on the content of this passenger protocol’s header since BGP FlowSpec does not provide the operator a way to address arbitrary locations within a packet.

3.3. Non-IP traffic

Not all traffic is forwarded as IP packets. Layer 2 services abound, including flavors of BGP-signaled Ethernet VPNs such as BGP-EVPN, BGP-VPLS, FEC 129 VPWS (LDP-signaled VPWS with BGP Auto-Discovery).

Ongoing efforts such as [I-D.ietf-idr-flowspec-l2vpn] offer one approach, which is to add layer 2 fields as additional match conditions. This may suffice if a filter needs to be applied only to layer 2, or only to layer 3 header fields.

4. Specification

We define a new FlowSpec component, Type TBD, named "Flexible Match Conditions".

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

The length is a one octet unsigned integer field that contains the length of the value field in octets.

The value field itself is encoded using offset-type, offset-value, pattern-type and pattern-value.

Encoding: <offset-type (4 bits), offset-value (2 octets), pattern-type (4 bits), pattern-value (variable)>

The value field is 0 padded for byte alignment.

4.1. Offset-type

The combination of offset-type and offset-value defines where the match should begin for the pattern-value. This document defines the following offset types:

Value	Offset Type
0	Layer 3 - IP Header
1	Layer 4 - IP Header Data
2	Payload - TCP/UDP Data

Offset Types

The offset-type 0 for ’layer 3’ is defined as the start of the IP header.

The offset-type 1 for ’layer 4’ is defined as the start of the data portion of the IP header after the IP options.

The offset-type 2 for ’payload’ is defined as start of the TCP or UDP data. For TCP, the offset-type payload represents the beginning of the TCP data after any TCP options. Note that Flow Specification NLRI using the Flexible Match Condition component with offset-type 2 will result in not matching the pattern value in this component in case of non-first fragmented packet or in case it is combined with component type 2 IP Protocol other than 6 (TCP) and 17 (UDP).

4.2. Offset-value

The offset-value is a 2 octets unsigned integer field defining the number of bytes to ignore in the packet from the offset-type to match the pattern value.

Examples:

- The combination of offset-type 0 (Layer 3) and offset-value 0 defines an offset at the very beginning of the IP header.

- The combination of offset-type 1 (Layer 4) and offset-value 2 defines an offset two bytes after the beginning of the data portion of the IP header (after any IP options). Example, in the case of a UDP packet, this offset defines the beginning of the destination port header field.

- The combination of offset-type 2 (Payload) and offset-value 10 defines an offset ten bytes after the beginning of the TCP/UDP data payload.

4.3. Pattern-type

The pattern-type defines how the pattern value is matched. The following pattern-types are defined:

Value	Pattern Type
0	Bitmask match
1	POSIX Regular expression (regex) string match
2	PCRE Regular expression (regex) string match

Pattern Types

Pattern-type 0 MUST be implemented.

Pattern-type 1 and 2 for regular expressions are typically dedicated to hardware-accelerated and software-only forwarding planes or appliances that may be able to filter on more complex criteria. There is a plethora of regular expression engines and their supported flavor. The two flavors introduced in this document are:

- o POSIX regular expression string match: This type refers to extended regular expression (ERE) as defined by [IEEE.1003-2.1992].

- o PCRE regular expression string match: This type refers to Perl compatible regular expression as defined by PCRE documentation [1].

4.4. Pattern-value

4.4.1. Bitmask match

If the pattern-type bitmask is selected, the pattern-value is encoded as {prefix, mask}, of equal length.

prefix - Provides a bit string to be matched. The prefix and mask fields are bitwise AND'ed to create a resulting pattern.

mask - Paired with the prefix field to create a bit string match. An unset bit is treated as a 'do not care' bit in the corresponding position in the prefix field. When a bit is set in the mask, the value of the bit in the corresponding location in the prefix field must match exactly.

4.4.2. Regular expression string match

If a regular expression pattern-type is selected, the pattern-value is encoded following the appropriate regular expression string match.

5. Flexible Match Conditions boundaries and additional considerations

The beginning of the match boundary is aligned with the FlowSpec AFI/SAFI to which the flexible match rule belongs. For instance, with FlowSpec for IPv4 traffic, the smallest offset can only start at the first bit of the IPv4 header.

The end of the match boundary MUST be the lesser of either the last bit in a packet or the Maximum Readable Length (Section 2) that a forwarding implementation can parse from a packet and make available for filtering. As the MRL will be implementation-dependent, it needs to be known to the Flow Specification controller. That can be communicated out-of-band via configuration or signaled using future BGP or IGP extensions.

The Maximum Pattern Length (Section 2) for the pattern-value can also be forwarding implementation dependant and may need to be known to the Flow Specification controller or communicated out-of-band.

It is not required that all nodes in a filtering domain have a common or minimum MRL and MPL. This does not remove the need for a Flow Specification controller to take MRL and MPL into account when creating flexible filters. This can be useful if the Flow Specification controller does not have direct BGP peering with all FlowSpec enforcers and may not receive a BGP Notification if it advertises a flexible match that exceeds the MRL or MPL of a given node.

6. Error Handling

Malicious, misbehaving, or misunderstanding implementations could advertise semantically incorrect values. Care must be taken to minimize fallout from attempting to parse such data. Any well-behaved implementation SHOULD verify that the minimum packet length undergoing a match equals (match from the offset + pattern-value length).

7. Security Considerations

This document introduces no additional security considerations beyond those already covered in [RFC5575] .

8. IANA Considerations

IANA is requested to assign a type from the First Come First Served range of the "Flow Spec Component Types" registry:

Type Value	Name	Reference
TBD	Flexible Match Conditions	this document

Reference: this document

Registry Owner/Change Controller: IESG

Registration procedures:

Range	Registration Procedures
0-127	IETF Review
128-249	First Come First Served
250-254	Experimental
255	Reserved

Note: a separate "owner" column is not provided because the owner of all registrations, once made, is "IESG".

9. Acknowledgements

We wish to thank John Scudder, Michael Gallagher, Ron Bonica, Jeff Haas, Sudipto Nandi, Brian St Pierre and Rafal Jan Szarecki for their valuable comments and suggestions on this document.

10. References

10.1. 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>.

[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>.

10.2. Informative References

[I-D.ietf-idr-flowspec-l2vpn] Weiguo, H., Eastlake, D., Litkowski, S., and S. Zhuang, "BGP Dissemination of L2 Flow Specification Rules", draft-ietf-idr-flowspec-l2vpn-16 (work in progress), November 2020.

[IEEE.1003-2.1992] Institute of Electrical and Electronics Engineers, "Information Technology - Portable Operating System Interface (POSIX) - Part 2: Shell and Utilities (Vol. 1)", IEEE Standard 1003.2, 1992.

[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>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

10.3. URIs

[1] https://www.pcre.org/original/pcre.txt

Authors' Addresses

Anurag Khare (editor)
Ciena Corporation

Email: ak@ciena.com

Philippe Bergeon (editor)
Nokia
600 March Road
Ottawa, Ontario K2K2E6
CA

Email: philippe.bergeon@nokia.com

Vijay Kestur
Juniper Networks, Inc.

Email: vkestur@juniper.net

Luay Jalil
Verizon

Email: luay.jalil@one.verizon.com

Kirill Kasavchenko

Email: kkasavchenko.mail@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 30, 2021

Z. Li
L. Li
Huawei
H. Chen
Futurewei
C. Loibl
Next Layer Communications
G. Mishra
Verizon Inc.
Y. Fan
Casa Systems
Y. Zhu
China Telecom
L. Liu
Fujitsu
X. Liu
Volta Networks

March 29, 2021

BGP Flow Specification for SRv6
draft-li-idr-flowspec-srv6-05

Abstract

This document proposes extensions to BGP Flow Specification for SRv6 for filtering SRv6 packets that match a sequence of conditions.

Requirements Language

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

Status of This Memo

This Internet-Draft will expire on September 30, 2021.

Copyright Notice

Table of Contents

1. Introduction

[RFC8955] describes in details about a new BGP NLRI to distribute a flow specification, which is an n-tuple comprising a sequence of matching criteria that can be applied to IP traffic. [RFC8956] extends [RFC8955] to make it also usable and applicable to IPv6 data packets. [I-D.ietf-idr-flowspec-l2vpn] extends the flow-spec rules for layer 2 Ethernet packets.

Segment Routing (SR) for unicast traffic has been proposed to cope with the usecases in traffic engineering, fast re-reroute, service chain, etc. SR architecture can be implemented over an IPv6 data plane using a new type of Segment Routing Header (SRH) [I-D.ietf-6man-segment-routing-header]. SRv6 Network Programming [RFC8986] defines the SRv6 network programming concept and its most basic functions. An SRv6 SID may have the form of LOC:FUNCT:ARGS::.

LOC: Each operator is free to use the locator length it chooses. Most often the LOC part of the SID is routable and leads to the node which instantiates that SID.

FUNCT: The FUNCT part of the SID is an opaque identification of a local function bound to the SID. (e.g. End: Endpoint, End.X, End.T, End.DX2 etc.).

ARGS: A function may require additional arguments that would be placed immediately after the FUNCT.

This document specifies two new BGP Flow Specification (FS) component types to support Segment Routing over IPv6 data plane (SRv6) filtering. The match field is destination address of IPv6 header, but it’s a SID copy from SRH rather than a traditional IPv6 address (refer to Figure 1).

```
            +-----------------------------+
 IPv6 Header|     SA      |     DA        |<--Match field of this document
            +--------------------^--------+
                                 |
            +--------------------|--------+
            |             +-------------+ |     +-------------------+
            |             | Segment[0]  +-------> Loc | Func | Args |
            |             +-------------+ |     +-------------------+
            |             | Segment[1]  | |
            |             +-------------+ |
            |             |     ...     | |
   SR Header|             +-------------+ |
            |             | Segment[n]  | |
            |             +-------------+ |
            |             +-------------+ |
            |             ~  Option TLV ~ |
            |             +-------------+ |
            +-----------------------------+
```

Figure 1: Match Field

2. Definitions and Acronyms

- FS: Flow Specification
- BGP-FS: Border Gateway Protocol (BGP) Flow Specification (FS)
- SR: Segment Routing
- SRH: SR Header.

- SRv6: IPv6 Segment Routing, SRv6 is a method of forwarding IPv6 packets on the network based on the concept of source routing.

- SID: Segment Identifier

- BSID: Binding SID

3. The Flow Specification Encoding for SRv6

The Flow Specification NLRI-type consists of several optional components, each of which begins with a type field (1 octet) followed by a variable length parameter. 13 component types are defined in [RFC8955] and [RFC8956] for IPv4 and IPv6. This document defines two new component types for SRv6.

3.1. Type TBD1 - Whole SID

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

Contains a list of {operator, value} pairs that are used to match the SID/binding SID or a range of whole SID.

The operator byte is encoded as:

```
  0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+---+
| e | a | 0 | 0 | 0 |lt |gt |eq |
+---+---+---+---+---+---+---+---+
```

Where:

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

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

0 - SHOULD be set to 0 on NLRI encoding, and MUST be ignored during decoding.

lt - less than comparison between data and value.

gt - greater than comparison between data and value.

eq - equality between data and value.

The bits lt, gt, and eq can be combined to match the SID or a range of SID (e.g. less than SID1 and greater than SID2).

The value field is encoded as:

```
 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
+---------------------------------------------------------------+
~                         SID(128bits)                          ~
+---------------------------------------------------------------+
```

The format of SID is described in [I-D.ietf-6man-segment-routing-header] and [RFC8986]

3.2. Type TBD2 - Some bits of SID

For some scenarios route policy with the whole 128 bits SID matching is too long and not necessary. [RFC8986] defines the format of SID is LOC:FUNCT:ARGS::. In some scenarios, traffic packets can just match Locator, Function ID, Argument or some combinations of these different fields rather than whole 128 bits SID. In order to match the Function ID (FUNCT), the Locator (LOC) needs to be examined and matched first. The new component type TBD2 defined below is for matching some bits of SID.

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

Contains a list of {operator, value} pairs that are used to match some bits of SID.

The operator byte is encoded as:

```
   0   1   2   3   4   5   6   7
 +---+---+---+---+---+---+---+---+
 | e | a | field type|lt |gt |eq |
 +---+---+---+---+---+---+---+---+
```

Where:

e and a are the same as defined in Section "Type TBD1 - Whole SID".

field type:

000 : SID’s LOC bits

001 : SID’s FUNCT bits

010 : SID’s LOC:FUNCT bits

011 : SID’s FUNCT:ARGS bits

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

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

eq - equality between data’ and value’.

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

The value field is encoded below as the lengths in bits of LOC, FUNCT and ARGS followed by the SID rounding up to bytes:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  LOC Length   | FUNCT Length  | ARGS Length |     SID         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                         SID(continue)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

LOC Length : 1-octet field indicating the length in bits of LOC in SID.

FUNCT Length : 1-octet field indicating the length in bits of FUNCT in SID.

ARGS Length : 1-octet field indicating the length in bits of ARGS in SID.

SID : the SID containing LOC, FUNCT and ARGS, and rounding up to bytes.

4. Security Considerations

No new security issues are introduced to the BGP protocol by this specification over the security considerations in [RFC8955] and [RFC8956].

5. IANA Considerations

This section complies with [RFC7153].

Under "Flow Spec IPv6 Component Types" registry, IANA is requested to assign the following values:

Value	Name	Reference
TBD1 (14)	Whole SID	This Document
TBD2 (15)	Some bits of SID	This Document

6. Acknowledgments

The authors would like to thank Joel Halpern, Shunwan Zhuang and Rainsword Wang for their valuable suggestions and comments on this draft.

7. References

7.1. 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>.

[RFC7153] Rosen, E. and Y. Rekhter, "IANA Registries for BGP Extended Communities", RFC 7153, DOI 10.17487/RFC7153, March 2014, <https://www.rfc-editor.org/info/rfc7153>.

[RFC8955] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", RFC 8955, DOI 10.17487/RFC8955, December 2020, <https://www.rfc-editor.org/info/rfc8955>.

[RFC8956] Loibl, C., Ed., Raszuk, R., Ed., and S. Hares, Ed., "Dissemination of Flow Specification Rules for IPv6", RFC 8956, DOI 10.17487/RFC8956, December 2020, <https://www.rfc-editor.org/info/rfc8956>.

7.2. Informative References

[I-D.ietf-6man-segment-routing-header]
Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-26 (work in progress), October 2019.

[I-D.ietf-idr-flowspec-l2vpn]
Weiguo, H., Eastlake, D., Litkowski, S., and S. Zhuang, "BGP Dissemination of L2 Flow Specification Rules", draft-ietf-idr-flowspec-l2vpn-16 (work in progress), November 2020.

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

Authors' Addresses

Zhenbin Li
Huawei
156 Beiqing Road
Beijing, 100095
P.R. China

Email: lizhenbin@huawei.com

Lei Li
Huawei
156 Beiqing Road
Beijing 100095
P.R. China

Email: lily.lilei@huawei.com

Huaimo Chen
Futurewei
Boston, MA
USA

Email: Huaimo.chen@futurewei.com

Christoph Loibl
Next Layer Communications
Mariahilfer Guertel 37/7
Vienna 1150
AT

Email: cl@tix.at

Gyan S. Mishra
Verizon Inc.
13101 Columbia Pike
Silver Spring MD 20904
USA

Phone: 301 502-1347
Email: gyan.s.mishra@verizon.com

Yanhe Fan
Casa Systems
USA

Email: yfan@casa-systems.com

Yongqing Zhu
China Telecom
109, West Zhongshan Road, Tianhe District
Guangzhou 510000
China

Email: zhuyq.gd@chinatelecom.cn

Lei Liu
Fujitsu

USA

Email: liulei.kddi@gmail.com

Xufeng Liu
Volta Networks

McLean, VA
USA

Email: xufeng.liu.ietf@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

Y. Wang
S. Zhuang
Y. Gu
Huawei
R. Pang
China Unicom

February 22, 2021

BGP Extension for Advertising In-situ Flow Information Telemetry (IFIT) Capabilities
draft-wang-idr-bgp-ifit-capabilities-02

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

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]. It can provide flow information on the entire forwarding path on a per-packet basis in real time.

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 Border Gateway Protocol (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 Tail Community

For IPv4 networks [RFC4360], this section defines a new type of BGP extended community called IPv4-Address-Specific IFIT Extended Community. The IPv4-Address-Specific IFIT Tail 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 TBA1.

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 (TBA1)|   Originating IPv4 Address    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |Originating IPv4 Address(cont.)|     IFIT Capabilities         |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2. IPv4-Address-Specific IFIT Tail Community

- Originating IPv4 Address field: A 4 octets field. A IPv4 address of the IFIT decapsulation node. It is an IPv4 unicast address assigned by one of the Internet registries

- IFIT Capabilities: A 2 octets field. as defined in previous setion.

4.2. IPv6-Address-Specific IFIT Tail Community

For IPv6 networks [RFC5701], this section defines a new type of BGP extended community called IPv6-Address-Specific IFIT Extended Community. The IPv6-Address-Specific IFIT Tail 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 TBA2.

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 (TBA2)|   Originating IPv6 Address    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               Originating IPv6 Address (cont.)                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               Originating IPv6 Address (cont.)                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               Originating IPv6 Address (cont.)                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Originating IPv6 Address(cont.)|     IFIT Capabilities         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3. IPv6-Address-Specific IFIT Tail Community

- o Originating IPv6 Address field: A IPv6 address of the IFIT decapsulation node. It is an IPv6 unicast address assigned by one of the Internet registries.

- o IFIT Capabilities: as defined in previous setion.

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 IFIT Next-Hop Capability is a triple (Capability Code, Capability Length, Capability Value) aka a 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Capability Code (TBA3)    |        Capability Length      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       IFIT Capabilities       | ORIG. IP Address(4/16 octets) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4. BGP Next-Hop Capability

- 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 TBA3.

- 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.

- IFIT Capabilities: as defined in previous setion.

- ORIG. IP Address: An IPv4 or IPv6 Address of the IFIT decapsulation node. It is an IPv4 or IPv6 unicast address assigned by one of the Internet registries.

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

The IANA is requested to make the assignments for IPv4-Address-Specific IFIT Tail Community and IPv6-Address-Specific IFIT Tail Community:

Value	Description	Reference
TBA1	IPv4-Address-Specific IFIT Tail Community	This document
TBA2	IPv6-Address-Specific IFIT Tail Community	This document

The IANA is requested to make the assignments for IFIT Next-Hop Capability:

Value	Description	Reference
TBA3	IFIT Capabilities	This document

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.

[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>.

[RFC5701] Rekhter, Y., "IPv6 Address Specific BGP Extended Community Attribute", RFC 5701, DOI 10.17487/RFC5701, November 2009, <https://www.rfc-editor.org/info/rfc5701>.

10.2. Informative References

[I-D.ietf-ippm-ioam-data] Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 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.

[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>.

[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>.

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

Ran Pang
China Unicom
Beijing
China

Email: pangran@chinaunicom.cn

IDR Working Group A. Wang
Internet-Draft W. Wang
Intended status: Informational China Telecom
Expires: September 9, 2021 G. Mishra
Verizon Inc.
H. Wang
S. Zhuang
J. Dong
Huawei Technologies
March 8, 2021

Analysis of VPN Routes Control in Shared BGP Session
draft-wang-idr-vpn-routes-control-analysis-03

Abstract

This draft analyzes some scenarios and the necessaries for VPN routes control in the shared BGP session, which can be the used as the base for the design of related solutions.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

BGP Maximum Prefix feature [RFC4486] is often used at the network boundary to control the number of prefixes to be injected into the network. But for some scenarios when the VPN routes from several VRFs are advertised via one shared BGP session, there is lack of appropriate methods to control the flooding of VPN routes within one VRF to overwhelm the process of VPN routes in other VRFs. That is to say, the excessive VPN routes advertisement should be controlled individually for each VRF in such shared BGP session.

The following sections analyzes the scenarios that are necessary to such mechanism.

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]

- o RR: Router Reflector, provides a simple solution to the problem of IBGP full mesh connection in large-scale IBGP implementation.

- o VRF: Virtual Routing Forwarding, a virtual routing table based on VPN instance.

4. Inter-AS VPN Option B/AB Scenario

For inter-AS VPN deployment option B/AB scenario, as described in Figure 1, there is one BGP session between ASBR1 and ASBR2, which is used to advertise the VPN routes from VPN1 and VPN2 VRF. Normally the operator will deploy the BGP maximum prefixes feature under different address families between the ASBR1 and ASBR2, but the threshold must be set very high to cope with the situation when all the VRFs in each family reach their VPN routes limit simultaneously. In case VPN routes in only one of VRF, for example VPN1 in PE3, advertises excess VPN routes(with RD set to RD31 and RT import/export set to RT1. Configurations on other PEs are similar) into the network, but VPN routes advertisement in other VRFs are in normal, the prefix bar set between the ASBRs will not take effect. Such excessive VPN routes will be advertised into the AS1, to PE1 and PE2 respectively.

PE1 in this example, provides the services for VPN2 at the same time. If it receives the excessive VPN routes for VPN1 from ASBR1, although such VPN routes have exceeded the limit within the VRF VPN1, it can’t break the BGP session with ASBR1 directly, because the VPN prefix limit is to prevent a flood from errors or other issues but does not prevent the device from being overwhelmed and resources exhausted.

All it can do is to receive and process the excessive BGP updates continuously, parse the excessive VPN routes for VPN1 and drop it, extract the VPN routes for VPN2 and install it.

Doing so can certainly influence the performance of PE1 to serve the other VPN services on it, considering that there are hundreds of VRFs deployed on it.

PE1 should have the capability to control the advertisement of specified excessive VPN routes from its BGP peer. The ASBR should also have such capability.

The excessive VPN routes may carry just one RT(for example in VPN1 on PE3), or carry more than one RTs(for example in VPN2 on PE3). Such excessive VPN routes may be imported into one VRF(for example VPN1 on PE1) or more than one VRFs(for example both VPN2 and VPN3 import the VPN routes with RD32, which has attached RT2 and RT3 together when they are advertised)

```
 +---------------------------+           +------------------------------+
 |                           |           |                              |
 |                           |           |                              |
 |   +---------+             |           |            +---------+       |
 |   |   PE1   |             |           |            |   PE3   |       |
 |   +---------+             |           |            +---------+       |
 |VPN1(RD11,RT1)\            |           |           /VPN1(RD31,RT1)    |
 |VPN2(RD12,RT2)  \+---------+  MP-EBGP  +---------+/ VPN2(RD32,RT2,RT3)|
 |VPN3(RD3 ,RT3)   |         |           |         |                    |
 |                 |  ASBR1  |-----------|  ASBR2  |                    |
 |                 |         |           |         |                    |
 |                 +---------+           +---------+                    |
 |                /          |           |         \                    |
 |   +---------+/            |           |           \+---------+       |
 |   |   PE2   |             |           |            |   PE4   |       |
 |   +---------+             |           |            +---------+       |
 |VPN1(RD21,RT1)             |           |           VPN2(RD42,RT2)     |
 |VPN2(RD2 ,RT2)             |           |           VPN3(RD3, RT3)     |
 |           AS1             |           |           AS2                |
 +---------------------------+           +------------------------------+
```

Figure 1: The Option B/Option AB cross-domain scenario

5. Inter-AS VPN Option C Scenario

For inter-AS VPN deployment option C scenario, as that described in Figure 2, there is one BGP session between RR1 and RR2, which is used to advertise the VPN routes from all the VRFs that located on the edge routers(PE1 and PE2). The BGP maximum prefix bar can't also prevent the excessive advertisement of VPN routes in one VRF, and such abnormal behavior in one VRF can certainly influence the performances of PEs to serve other normal VRFs.

PE and RR should all have some capabilities to control the specified excessive VPN routes to be advertised from its upstream BGP peer.

```
                                    MP-EBGP
                +------------------------------------------+
                |                                          |
   +------------+--------------+              +------------+------------------+
   |        +---+----+         |              |            +----+----+        |
   |    +---+  RR1   +---+     |              |       +----+   RR2   +---+    |
   |    |   +--------+   |     |              |       |    +---------+   |    |
   |    |                |     |              |       |                  |    |
   |    |IBGP        IBGP|     |              |       |IBGP          IBGP|    |
   |    |                |     |              |       |                  |    |
   | +--+---+        +---+--+  |              |   +---+---+           +--+--+ |
   | |  PE1 |        | ASBR1|--|--------------|---| ASBR2 |           | PE2 | |
   | +------+        +------+  |              |   +-------+           +-----+ |
   |           AS1             |              |           AS2                 |
   +---------------------------+              +-------------------------------+
```

Figure 2: The Option C cross-domain scenario

6. Intra-AS VPN RR Deployment Scenario

For intra-AS VPN deployment, as depicted in Figure 3, if the RR is present, the above excess VPN routes advertisement churn can also occurs. For example, if PE3 receives excessive VPN routes for VPN1 VRF(there may be several reasons for this to occur, for example, multiple CEs connect to PE3 advertising routes simultaneously causing a wave of routes, redistribution from VRF to VRF, or from GRT to VRF on PE3 etc.), it will advertise such excessive VPN routes to RR and then to PE1. The BGP session between RR and PE3, and the BGP session between RR and PE1 can’t prevent this to occur.

The RD in each VPN may be allocated and unique for each VPN on each PE(as example VPN1 in Figure 3), or only unique for each VPN(as example VPN2 in Figure 3).

Each VPN may be associated with one or more RTs. The excessive VPN routes may have only one RT(for example, the excessive VPN routes from PE3 has the RD equal to RD31 and RT is set only to RT1)

When PE1 in this figure receives such excessive VPN routes, it can only process them, among the other normal BGP updates. This can certainly influence process of VPN routes for other normal services, the consequences on the receiving PE1 may be the one or more of the followings:

a) PE1 can’t process a given number of routes in time period X leading to dropping of routes

b) Delayed processing that may result in an incomplete number of inputs to the BGP Best Path decision.

c) L3VPN customers experiencing an incorrect VPN specification for some time period Y.

d) The convergence of control plane processing impacts the traffic forwarding

PE and RR should all have some capabilities to control the specified excessive VPN routes to be advertised from its upstream BGP peer.

```
          +-----------------------------------------------+
          |   +---------+                  +---------+    |
          |   |   PE1   |                  |   PE4   |    |
          |   +---------+                  +---------+    |
          | VPN1(RD11,RT1) \              / VPN2(RD12,RT2)|
          | VPN2(RD12,RT2)  \+---------+ /                |
          |                 |         |                   |
          |                 |   RR    |                   |
          |                 |         |                   |
          |                 +---------+ \                 |
          |                /              \               |
          |   +---------+/                 +---------+    |
          |   |   PE2   |                  |   PE3   |    |
          |   +---------+                  +---------+    |
          | VPN1(RD21,RT1)              VPN1(RD31,RT1,RT2)|
          |                             VPN2(RD12,RT2)    |
          |                                               |
          |                    AS100                      |
          +-----------------------------------------------+
```

Figure 3: Intra-AS VPN RR deployment scenario

7. VPN Routes Shared on one PE

The scenarios described above are mainly in device level, that is to say, if the receiving PE has some mechanism to control the excess VPN routes advertisement from its BGP neighbor, the failure churn effect can be controlled then. But there are also situations that the granular control should be took place within the receiving PE itself.

Figure 4 below describes such scenario. There are four VRFs on PE, and three of them import the same VPN routes that carry route target RT3. Such deployment can occur in the inter-VRF communication scenario. If the threshold of VPN route-limit for these VRFs is set different, for example, are max-vpn-routes-vrf1, max-vpn-routes-vrf2, max-vpn-routes-vrf3, max-vpn-routes-vrf4 respectively, and these values have the following order, as max-vpn-routes-vrf1<max-vpn-routes-vrf2< max-vpn-routes-vrf3<max-vpn-routes-vrf4.

If the VPN routes that associates with RT3 is overwhelming, the VRF1 will reach its maximum VPN threshold first. At such stage, the PE device can't send the control message to its BGP neighbor on behalf of all the VRFs on it, because other VRFs have still the desire to receive such VPN routes and have the capacities to store them.

In such situation, the PE device should have some mechanisms to control the distribution of global VPN routes to its individual VRF table. Only when all of VRFs on it don't want some VPN routes, then the PE device can send the VPN routes filter control message to its BGP neighbor (RR in this example).

```
+------------------------------------------------+
|      +--------+                                |
|      |   RR   |--------------------+           |
|      +--------+                    |           |
|          |                         |           |
|          |                         |           |
|      +--------+                +---+---+       |
|      |   PE1  |                |  PE2  |       |
|      +--------+                +-------+       |
|   VPN1(RD21,RT1,RT3)       VPN1(RD21,RT1)      |
|   VPN2(RD22,RT2)           VPN3(RD23,RT3)      |
|   VPN3(RD23,RT3)                               |
|                                                |
|                     AS 100                     |
+------------------------------------------------+
```

Figure 4: The scenario of several VRFs in a PE import VPN routes carries the same RT

.

8. Requirements for the solutions

Based on the above scenarios description, the potential solutions should meet the following requirements:

a) The control message for the specified VPN routes should be triggered automatically upon the excessive VPN routes reach its limit.

b) The control message should be sent only out the device when all the VRFs on it can't or don't want to process it, or the process of such excessive routes has exceed its own capability.

c) For RR devices, such control message should be only flooded to its upstream BGP neighbor when all its clients can’t or don’t want to process it, or the process of such excessive routes has exceed its own capability.

d) For ASBR devices, such control message should be only flooded to its upstream BGP neighbor when all its downstream BGP peers can’t or don’t want to process it, or the process of such excessive routes has exceed its own capability.

e) The trigger and removal of such control message should avoid the possible flapping of excessive VPN routes advertisement.

9. Security Considerations

TBD.

10. IANA Considerations

This document requires no IANA considerations.

11. Acknowledgement

Thanks Robert Raszuk, Jim Uttaro, Jakob Heitz, Shuanglong Chen for their valuable comments and discussions of scenarios described in this draft.

12. 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>.

[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>.

[RFC4486] Chen, E. and V. Gillet, "Subcodes for BGP Cease Notification Message", RFC 4486, DOI 10.17487/RFC4486, April 2006, <https://www.rfc-editor.org/info/rfc4486>.

Authors’ Addresses

Aijun Wang
China Telecom
Beiqijia Town, Changping District
Beijing, Beijing 102209
China

Email: wangaj3@chinatelecom.cn

Wei Wang
China Telecom
Beiqijia Town, Changping District
Beijing, Beijing 102209
China

Email: wangw36@chinatelecom.cn

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

Haibo Wang
Huawei Technologies
Huawei Building, No.156 Beiqing Rd.
Beijing, Beijing 100095
China

Email: rainsword.wang@huawei.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

IDR Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

Yao. Liu
Shaofu. Peng
ZTE Corp.

February 22, 2021

BGP Extension for SR-MPLS Entropy Label Position
draft-zhou-idr-bgp-srmpls-elp-02

Abstract

This document proposes extensions for BGP to indicate the entropy label position in the SR-MPLS label stack when distributing SR Policy candidate paths.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Segment Routing (SR) leverages the source routing paradigm. Segment Routing can be instantiated on MPLS data plane which is referred to as SR-MPLS [RFC8660]. SR-MPLS leverages the MPLS label stack to construct the SR path.

Entropy labels (ELs) [RFC6790] are used in the MPLS data plane to provide entropy for load-balancing. The idea behind the entropy label is that the ingress router computes a hash based on several fields from a given packet and places the result in an additional label named "entropy label". Then, this entropy label can be used as part of the hash keys used by an LSR. Using the entropy label as part of the hash keys reduces the need for deep packet inspection in the LSR while keeping a good level of entropy in the load-balancing.

[RFC8662] proposes to use entropy labels for SR-MPLS networks and mutiple < ELI, EL> pairs may be inserted in the SR-MPLS label stack. The ingress node may decide the number and position of the ELI/ELs which need to be inserted into the label stack, that is termed as ELP (Entropy Label Position). In some cases, centralized controllers are used to perform the TE path computation for intra or inter-domain scenarios, thus it is also the responsibility of the controller to calculate ELP and inform it to the headend of the SR-TE path.

[I-D.ietf-idr-segment-routing-te-policy] defines the specific process of how the controller/PCE in the SR network passes the path calculation result of the SR-TE policy to the headend of the network through BGP.

This document proposes extensions for BGP to indicate the ELP in the segment list when distribute SR Policy candidate paths using BGP for SR-MPLS networks.

2. Conventions used in this document

2.1. 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.

2.2. Terminology and Acronyms

EL: Entropy Label

ELI: Entropy Label Indicator

ELC: Entropy Label Capability

ERLD: Entropy Readable Label Depth

ELP: Entropy Label Position

MSD: Maximum SID Depth

3. Entropy Labels in SR-MPLS Scenario with a Controller

[RFC8662] proposes to use entropy labels for SR-MPLS networks. The Entropy Readable Label Depth (ERLD) is defined as the number of labels which means that the router will perform load-balancing using the ELI/EL. An appropriate algorithm should consider the following criteria:

- a limited number of < ELI, EL> pairs SHOULD be inserted in the SR-MPLS label stack;

- the inserted positions SHOULD be whithin the ERLD of a maximize number of transit LSRs;

- a minimum number of < ELI, EL> pairs SHOULD be inserted while satisfying the above criteria.

As shown in Figure 1, in SR-MPLS inter-domain scenario, an path from A to Z is required, a centralized controller performs the computation of the end-to-end path, along which traffic load-balancing is required. The controller can also be used to perform the computation of the the Entropy Label Position (ELP) of the segment list that corresponds to the path. The ELP includs the number and the position of the ELI/ELs.

Multiple limitations MUST be taken into account by the controller during path calculation, including Entropy Readable Label Depth (ERLD) and Maximum SID Depth (MSD) etc. The ERLD is defined as the number of labels which means that the node will perform load-balancing using the ELI/EL pairs. And each ELI/EL pair must be within the ERLD of the node. The Maximum SID Depth (MSD) defines the maximum number of any kind of labels(service, entropy, transport, etc.) and it is a limit when the ingress node imposing ELI/EL pairs on the SR label stack.

The controller can get ERLD value and MSD via protocols. The ERLD value can be advertised via IS-IS[I-D.ietf-isis-mpls-elc], OSPF[I-D.ietf-ospf-mpls-elc], BGP-LS[I-D.ietf-idr-bgp-ls-segment-routing-ext]. [RFC8476], [RFC8491], and[RFC8814] provide examples of advertisement of the MSD.

As specified in section 7.2 [RFC8662], other criteria includes segment type, preference for a part of the path, and etc. It’s a matter of implementation.

```
 ....................   ....................    ......................
 .                  .   .                  .    .                    .
 .+---+       +---+ .   . +---+      +---+ .    .+---+      +----+   .
 .| A |-------| B |------ | C |------| X |-------| Y |------| Z  |   .
 .+---+       +---+ .   . +---+      +---+ .    .+---+      +----+   .
 .     domain 1     .   .      domain 2    .    .     domain 3       .
 ....................   ....................    ......................
```

Figure 1: Entropy Labels in SR-MPLS Inter-Domain Scenario

4. BGP Extensions

The Segment Flags for Segment Sub-TLVs are defined in Section 2.4.4.2.12 of [I-D.ietf-idr-segment-routing-te-policy]. In this document, the ELP information is transmitted by extending the flags of Segment Sub-TLVs.

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

E-Flag: This flag, when set, indicates that presence of < ELI, EL> label pairs which are inserted after this segment. E-Flag is applicable to Segment Types A, C, D, E, F, G and H. If E-Flag appears with Segment Types B, I, J and K, it MUST be ignored. .

5. Operations

Node A receives an SR Policy NLRI with an Segment List sub-TLV from the controller. The Segment List sub-TLV contains multiple Segment sub-TLVs, for example, <S1, S2, S3, S4, S5, S6>, the E-Flags of S3 and S6 are set, it indicates that if load-balancing is required, two <ELI, EL> pairs SHOULD be inserted into the label stack of the SR-TE forwarding entry, respectively after the Label for S3 and Label for S6.

The value of EL is supplemented by the ingress node according to load-balancing function of the appropriate keys extracted from a given packet. After inserting ELI/ELs, the label stack on the ingress node would be <label1, label2, label3, ELI, EL, label4, label5, label6, ELI, EL>.

6. Security Considerations

TBD

7. IANA Considerations

This document requests bit 4 for Entropy Label Flag.

Bit	Description	Reference
4	Entropy Label Position Flag(E-Flag)	This document

8. References

8.1. 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-11 (work in progress), November 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>.

[RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <https://www.rfc-editor.org/info/rfc6790>.

[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>.

[RFC8662] Kini, S., Kompella, K., Sivabalan, S., Litkowski, S., Shakir, R., and J. Tantsura, "Entropy Label for Source Packet Routing in Networking (SPRING) Tunnels", RFC 8662, DOI 10.17487/RFC8662, December 2019, <https://www.rfc-editor.org/info/rfc8662>.

8.2. Informative References

[I-D.ietf-idr-bgp-ls-segment-routing-ext]
Previdi, S., Talaulikar, K., Filsfils, C., Gredler, H., and M. Chen, "BGP Link-State extensions for Segment Routing", draft-ietf-idr-bgp-ls-segment-routing-ext-16 (work in progress), June 2019.

[I-D.ietf-isis-mpls-elc]
Xu, X., Kini, S., Psenak, P., Filsfils, C., Litkowski, S., and M. Bocci, "Signaling Entropy Label Capability and Entropy Readable Label Depth Using IS-IS", draft-ietf-isis-mpls-elc-13 (work in progress), May 2020.

[I-D.ietf-ospf-mpls-elc]
Xu, X., Kini, S., Psenak, P., Filsfils, C., Litkowski, S., and M. Bocci, "Signaling Entropy Label Capability and Entropy Readable Label Depth Using OSPF", draft-ietf-ospf-mpls-elc-15 (work in progress), June 2020.

[RFC8476] Tantsura, J., Chunduri, U., Aldrin, S., and P. Psenak, "Signaling Maximum SID Depth (MSD) Using OSPF", RFC 8476, DOI 10.17487/RFC8476, December 2018, <https://www.rfc-editor.org/info/rfc8476>.

[RFC8491] Tantsura, J., Chunduri, U., Aldrin, S., and L. Ginsberg, "Signaling Maximum SID Depth (MSD) Using IS-IS", RFC 8491, DOI 10.17487/RFC8491, November 2018, <https://www.rfc-editor.org/info/rfc8491>.

[RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", RFC 8660, DOI 10.17487/RFC8660, December 2019, <https://www.rfc-editor.org/info/rfc8660>.

[RFC8814] Tantsura, J., Chunduri, U., Talaulikar, K., Mirsky, G., and N. Triantafillis, "Signaling Maximum SID Depth (MSD) Using the Border Gateway Protocol - Link State", RFC 8814, DOI 10.17487/RFC8814, August 2020, <https://www.rfc-editor.org/info/rfc8814>.

Authors' Addresses

Liu Yao
ZTE Corp.

Email: liu.yao71@zte.com.cn

Peng Shaofu
ZTE Corp.

Email: peng.shaofu@zte.com.cn