BESS
Internet-Draft
Intended status: Standards Track
Expires: February 21, 2021

Z. Zhang
Juniper Networks
R. Raszuk
Bloomberg LP
D. Pacella
Verizon
A. Gulko
Refinitiv

August 20, 2020

Controller Based BGP Multicast Signaling
draft-ietf-bess-bgp-multicast-controller-04

Abstract

This document specifies a way that one or more centralized controllers can use BGP to set up a multicast distribution tree in a network. In the case of labeled tree, the labels are assigned by the controllers either from the controllers' local label spaces, or from a common Segment Routing Global Block (SRGB), or from each routers Segment Routing Local Block (SRLB) that the controllers learn. In case of labeled unidirectional tree and label allocation from the common SRGB or from the controllers' local spaces, a single common label can be used for all routers on the tree to send and receive traffic with. Since the controllers calculate the trees, they can use sophisticated algorithms and constraints to achieve traffic engineering.

Requirements Language

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

Status of This Memo

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

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

Copyright Notice

Table of Contents

1. Overview

1.1. Introduction

[I-D.ietf-bess-bgp-multicast] describes a way to use BGP as a replacement signaling for PIM [RFC7761] or mLDP [RFC6388]. The BGP-based multicast signaling described there provides a mechanism for setting up both (s,g)/(*,g) multicast trees (as PIM does, but optionally with labels) and labeled (MPLS) multicast tunnels (as mLDP does). Each router on a tree performs essentially the same procedures as it would perform if using PIM or mLDP, but all the inter-router signaling is done using BGP.

These procedures allow the routers to set up a separate tree for each individual multicast (x,g) flow where the 'x' could be either 's' or '*', but they also allow the routers to set up trees that are used for more than one flow. In the latter case, the trees are often referred to as "multicast tunnels" or "multipoint tunnels", and specifically in this document they are mLDP tunnels (except that they are set up with BGP signaling). While it actually does not have to be restricted to mLDP tunnels, mLDP FEC is conveniently borrowed to identify the tunnel. In the rest of the document, the term tree and tunnel are used interchangeably.

The trees/tunnels are set up using the "receiver-initiated join" technique of PIM/mLDP, hop by hop from downstream routers towards the root. The BGP messages are either sent hop by hop between downstream routers and their upstream neighbors, or can be reflected by Route Reflectors (RRs).

As an alternative to each hop independently determining its upstream router and signaling upstream towards the root (following PIM/mLDP model), the entire tree can be calculated by a centralized controller, and the signaling can be entirely done from the controller, using the same BGP messages as defined in [I-D.ietf-bess-bgp-multicast]. For that, some additional procedures and optimizations are specified in this document.

While it is outside the scope of this document, signaling from the controllers could be done via other means as well, like Netconf or any other SDN methods.

1.2. Resilience

Each router could establish direct BGP sessions with one or more controllers, or it could establish BGP sessions with RRs who in turn peer with controllers. For the same tree/tunnel, each controller may independently calculate the tree/tunnel and signal the routers on the tree/tunnel using MCAST-TREE S-PMSI/Leaf A-D routes [I-D.ietf-bess-bgp-multicast]. How the tree/tunnel roots/leaves are discovered and how the calculation is done are outside the scope of this document.

On each router, BGP route selection rules will lead to one controller's route for the tree/tunnel being selected as the active route and used for setting up forwarding state. As long as all the routers on a tree/tunnel consistently pick the same controller's routes for the tree/tunnel, the setup should be consistent. If the tree/tunnel is labeled, different labels will be used from different controllers so there is no traffic loop issue even if the routers do not consistently select the same controlle's routes. In the unlabeled case, to ensure the consistency the selection SHOULD be solely based on the identifier of the controller, which could be carried in an Address Specific Extended Community (EC).

Another consistency issue is when a bidirectional tree/tunnel needs to be re-routed. Because this is no longer triggered hop-by-hop from downstream to upstream, it is possible that the upstream change happens before the downstream, causing traffic loop. In the unlabeled case, there is no good solution (other than that the controller issues upstream change only after it gets acknowledgement from downstream). In the labeled case, as long as a new label is used there should be no problem.

Besides the traffic loop issue, there could be transient traffic loss before both the upstream and downstream's forwarding state are updated. This could be mitigated if the upstream keep sending traffic on the old path (in addition to the new path) and the downstream keep accepting traffic on the old path (but not on the new path) for some time. It is a local matter when for the downstream to switch to the new path - it could be data driven (e.g., after traffic arrives on the new path) or timer driven.

For each tree, multiple disjoint instances could be calculated and signaled for live-live protection. Different labels are used for different instances, so that the leaves can differentiate incoming

traffic on different instances. As far as transit routers are concerned, the instances are just independent. Note that the two instances are not expected to share common transit routers (it is otherwise outside the scope of this document/revision).

1.3. Signaling

Each router only receives Leaf A-D routes from the controllers but does not originate or re-advertise S-PMSI/Leaf A-D routes. The re-advertisement of a received route can be blocked based on the fact that a configured import RT matches the RT of the route, which indicates that this router is the target and consumer of the route hence it should not be re-advertised further. The routes includes the forwarding information in the form of Tunnel Encapsulation Attributes (TEA) [I-D.ietf-idr-tunnel-encaps], with enhancements specified in this document.

Suppose that for a particular tree, there are two downstream routers D1 and D2 for a particular upstream router U. A controller C may send two Leaf A-D routes to U, as if the two routes were originated by D1 and D2 but reflected by the controller. Alternatively, C could just send one route to U, with the Upstream Router's IP Address field set to U's IP address and the TEA specifying both the two downstreams and its upstream (see Section 2.1.4). In this case, the Originating Router's Address field of the Leaf A-D route is set to the controller's address. Note that for a TEA attached to a unicast NLRI, only one of the tunnels in a TEA is used for forwarding a particular packet, while all the tunnels in a TEA are used to reach multiple endpoints when it is attached to a multicast NLRI.

Note that, in case of labeled trees, the (x,g) or mLDP FEC signaling is actually not needed to transit routers but only needed on tunnel root/leaves. However, for consistency, the same signaling is used to all routers.

1.4. Label Allocation

In the case of labeled multicast signaled hop by hop towards the root, whether it's (x,g) multicast or "mLDP" tunnel, labels are assigned by a downstream router and advertised to its upstream router (from traffic direction point of view). In the case of controller based signaling, routers do not originate tree join (S-PMSI/Leaf A-D) routes anymore, so the controllers have to assign labels on behalf of routers, and there are three options for label assignment:

- From each router's SRLB that the controller learns

- From the common SRGB that the controller learns

o From the controller's local label space

Assignment from each router's SRLB is no different from each router assigning labels from its own local label space in the hop-by-hop signaling case. The assignments for a router is independent of assignments for another router, even for the same tree.

Assignment from the controller's local label space is upstream-assigned [RFC5331]. It is used if the controller does not learn the common SRGB or each router's SRLB. Assignment from the SRGB [RFC8402] is only meaningful if all SRGBs are the same and a single common label is used for all the routers on a tree in case of unidirectional tree/tunnel (Section 1.4.1). Otherwise, assignment from SRLB is preferred.

The choice of which of the options to use depends on many factors. An operator may want to use a single common label per tree for ease of monitoring and debugging, but that requires explicit RPF checking and either SRGB or upstream assigned labels, which may not be supported due to either the software or hardware limitations (e.g. label imposition/disposition limits). In an SR network, assignment from the common SRGB if it's required to use a single common label per unidirectional tree, or otherwise assignment from SRLB is a good choice because it does not require support for context label spaces.

1.4.1. Using a Common per-tree Label for All Routers

MPLS labels only have local significance. For an LSP that goes through a series of routers, each router allocates a label independently and it swaps the incoming label (that it advertised to its upstream) to an outgoing label (that it received from its downstream) when it forwards a labeled packet. Even if the incoming and outgoing labels happen to be the same on a particular router, that is just incidental.

With Segment Routing, it is becoming a common practice that all routers use the same SRGB so that a SID maps to the same label on all routers. This makes it easier for operators to monitor and debug their network. The same concept applies to multicast trees as well - a common per-tree label is used for a router to receive traffic from its upstream neighbor and replicate traffic to all its downstream neighbor.

However, a common per-tree label can only be used for unidirectional trees. Additionally, it requires each router to do explicit RPF check, so that only packets from its expected upstream neighbor are accepted. Otherwise, traffic loop may form during topology changes, because the forwarding state update is no longer ordered.

Traditionally, p2mp mpls forwarding does not require explicit RPF check as a downstream router advertises a label only to its upstream router and all traffic with that incoming label is presumed to be from the upstream router and accepted. When a downstream router switches to a different upstream router a different label will be advertised, so it can determine if traffic is from its expected upstream neighbor purely based on the label. Now with a single common label used for all routers on a tree to send and receive traffic with, a router can no longer determine if the traffic is from its expected neighbor just based on that common tree label. Therefore, explicit RPF check is needed. Instead of interface based RPF checking as in PIM case, neighbor based RPF checking is used - a label identifying the upstream neighbor precedes the tree label and the receiving router checks if that preceding neighbor label matches its expected upstream neighbor. Notice that this is similar to what's described in Section "9.1.1 Discarding Packets from Wrong PE" of RFC 6513 (an egress PE discards traffic sent from a wrong ingress PE). The only difference is one is used for label based forwarding and the other is used for (s,g) based forwarding. [note: for bidirectional trees, we may be able to use two labels per tree - one for upstream traffic and one for downstream traffic. This needs further verification].

Both the common per-tree label and the neighbor label are allocated either from the common SRGB or from the controller's local label space. In the latter case, an additional label identifying the controller's label space is needed, as described in the following section.

1.4.2. Upstream-assignment from Controller's Local Label Space

In this case in the multicast packet's label stack the tree label and upstream neighbor label (if used in case of single common-label per tree) are preceded by a downstream-assigned "context label". The context label identifies a context-specific label space (the controller's local label space), and the upstream-assigned label that follows it is looked up in that space.

This specification requires that, in case of upstream-assignment from a controller's local label space, each router D to assign, corresponding to each controller C, a context label that identifies the upstream-assigned label space used by that controller. This label, call it Lc-D, is communicated by D to C.

Suppose a controller is setting up unidirectional tree T. It assigns that tree the label Lt, and assigns label Lu to identify router U which is the upstream of router D on tree T. C needs to tell U: "to send a packet on the given tree/tunnel, one of the things you have to

do is push Lt onto the packet's label stack, then push Lu, then push Lc-D onto the packet's label stack, then unicast the packet to D". Controller C also needs to inform router D of the correspondence between <Lc-D, Lu, Lt> and tree T.

To achieve that, when C sends an S-PMSI/Leaf A-D route, for each tunnel in the TEA, it includes a label stack Sub-TLV [I-D.ietf-idr-tunnel-encaps], with the outer label being the context label Lc-D (received by the controller from the corresponding downstream), the next label being the upstream neighbor label Lu, and the inner label being the label Lt assigned by the controller for the tree. The router receiving the route will use the label stacks to send traffic to its downstreams.

For C to signal the expected label stack for D to receive traffic with, we overload a tunnel TLV in the TEA of the Leaf A-D route sent to D - if the tunnel TLV has a RPF sub-TLV (Section 2.1.4), then it indicates that this is actually for receiving traffic from the upstream.

1.5. Determining Root/Leaves

For the controller to calculate a tree, it needs to determine the root and leaves of the tree. This may be based on provisioning (static or dynamically programmed), or based on BGP signaling using the BGP multicast messages defined in [I-D.ietf-bess-bgp-multicast], as described in the following two sections.

In both cases, the BGP updates are targeted at the controller, via an address specific Route Target with Global Administration Field set to the controller's address and the Local Administration Field set to 0, or a value pre-assigned to identify a VPN.

1.5.1. PIM-SSM/Bidir or mLDP

In this case, the PIM Last Hop Routers (LHRs) with interested receivers or mLDP tunnel leaves encode a Leaf A-D route with the Upstream Router's IP Address field set to the controller's address and the Originating Router's IP Address set to the address of the LHR or the P2MP tunnel leaf. The encoded PIM SSM source or mLDP FEC provides root information and the Originating Router's IP Address provides leaves information.

1.5.2. PIM ASM

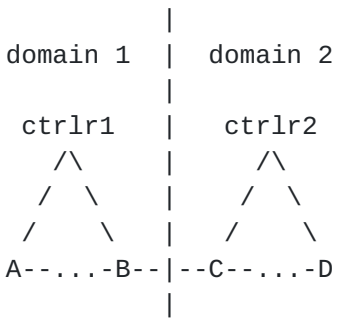
In this case, the First Hop Routers (FHRs) originate Source Active routes which provides root information, and the LHRs originate Leaf

A-D routes, encoded as in the PIM-SSM case except that it is (*,G) instead of (S,G). The Leaf A-D routes provide leaf information.

1.6. Multiple Domains

An end to end multicast tree may span multiple routing domains, and the setup of the tree in each domain may be done differently as specified in [I-D.ietf-bess-bgp-multicast]. This section discusses a few aspects specific to controller signaling.

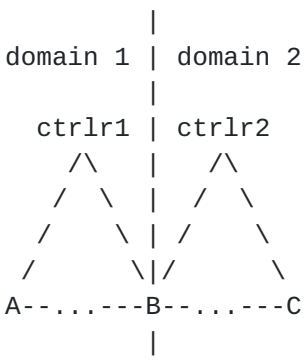
Consider two adjacent domains each with its own controller in the following configuration where router B is an upstream node of C for a multicast tree:



In the case of native (un-labeled) IP multicast, nothing special is needed. Controller 1 signals B to send traffic out of B-C link while Controller 2 signals C to accept traffic on the B-C link.

In the case of labeled IP multicast or mLDP tunnel, the controllers may be able to coordinate their actions such that Controller 1 signals B to send traffic out of B-C link with label X while Controller 2 signals C to accept traffic with the same label X on the B-C link. If the coordination is not possible, then C needs to use hop-by-hop BGP signaling to signal towards B, as specified in [I-D.ietf-bess-bgp-multicast].

The configuration could also be as following, where router B borders both domain 1 and domain 2 and is controlled by both controllers:



As discussed in Section 1.2, when B receives signaling from both Controller 1 and Controller 2, only one of the routes would be selected as the best route and used for programming the forwarding state of the corresponding segment. For B to stitch the two segments together, it is expected for B to know by provisioning that it is a border router so that B will look for the other segment (represented by the signaling from the other controller) and stitch the two together.

1.7. SR-P2MP

[I-D.voyer-pim-sr-p2mp-policy] describes an architecture to construct a Point-to-Multipoint (P2MP) tree to deliver Multi-point services in a Segment Routing domain. An SR P2MP tree is constructed by stitching together a set of Replication Segments that are specified in [I-D.voyer-spring-sr-replication-segment]. An SR Point-to-Multipoint (SR P2MP) Policy is used to define and instantiate a P2MP tree which is computed by a controller.

An SR P2MP tree is no different from an mLDP tunnel in MPLS forwarding plane. The difference is in control plane - instead of hop-by-hop mLDP signaling from leaves towards the root, to set up SR P2MP trees controllers program forwarding state (referred to as Replication Segments) to the root, leaves, and intermediate replication points using Netconf, PCEP, BGP or any other reasonable signaling/programming methods.

Procedures in this document can be used for controllers to set up SR P2MP trees with just an additional S-PMSI route type.

If/once the SR Replication Segment is extended to bi-redirectional, and SR MP2MP is introduced, the same procedures in this document would apply to SR MP2MP as well.

2. Specification

2.1. Enhancements to TEA

This document specifies two new Tunnel Types and new sub-TLV. The type codes will be assigned by IANA from the "BGP Tunnel Encapsulation Attribute Tunnel Types".

2.1.1. Any-Encapsulation Tunnel

When a multicast packet needs to be sent from an upstream node to a downstream node, it may not matter how it is sent - natively when the two nodes are directly connected or tunneled otherwise. In case of tunneling, it may not matter what kind of tunnel is used - MPLS, GRE, IPinIP, or whatever.

To support this, an "Any-Encapsulation" tunnel type is defined. This tunnel MUST have a Tunnel Endpoint Sub-TLV and SHOULD NOT have any other Sub-TLVs. The Tunnel Endpoint Sub-TLV specifies an IP address, which could be any of the following:

- An interface's local address - when a packet needs to sent out of the corresponding interface natively.
- An interface's remote address - when a packet needs to sent to the address natively.
- An address that is not directly connected - when a packet needs to be tunneled to the address (any tunnel type/instance can be used).

2.1.2. Load-balancing Tunnel

Consider that a multicast packet needs to be sent to a downstream node, which could be reached via four paths P1~P4. If it does not matter which of path is taken, an "Any-Encapsulation" tunnel with the Tunnel Endpoint Sub-TLV specifying the downstream node's loopback address works well. If the controller wants to specify that only P1~P2 should be used, then a "Load-balancing" tunnel needs to be used, listing P1 and P2 as member tunnels of the "Load-balancing" tunnel.

A load-balancing tunnel has one "Member Tunnels" Sub-TLV defined in this document. The Sub-TLV is a list of tunnels, each specifying a way to reach the downstream. A packet will be sent out of one of the tunnels listed in the Member Tunnels Sub-TLV of the load-balancing tunnel.

2.1.3. Receiving MPLS Label Stack

While [I-D.ietf-bess-bgp-multicast] uses S-PMSI A-D routes to signal forwarding information for MP2MP upstream traffic, when controller signaling is used, a single Leaf A-D route is used for both upstream and downstream traffic. Since different upstream and downstream labels need to be used, a new "Receiving MPLS Label Stack" of type TBD is added as a tunnel sub-TLV in addition to the existing MPLS Label Stack sub-TLV. Other than type difference, the two are the encoded the same way.

The Receiving MPLS Label Stack sub-TLV is added to each downstream tunnel in the TEA of Leaf A-D route for an MP2MP tunnel to specify the forwarding information for upstream traffic from the corresponding downstream node. A label stack instead of a single label is used because of the need for neighbor based RPF check, as further explained in the following section.

The Receiving MPLS Label Stack sub-TLV is also used for downstream traffic from the upstream for both P2MP and MP2MP, as specified below.

2.1.4. RPF Sub-TLV

The RPF sub-TLV has a type to be allocated by IANA and a one-octet length. The length is 0 currently, but if necessary in the future, sub-sub-TLVs could be placed in its value part. If the RPF sub-TLV appears in a tunnel, it indicates that the "tunnel" is for the upstream node instead of a downstream node. It contains an Receiving MPLS Label Stack sub-TLV for downstream traffic from the upstream node, and in case of MP2MP it also contains a regular MPLS Label Stack sub-TLV for upstream traffic to the upstream node.

The inner most label in the Receiving MPLS Label Stack is the incoming label identifying the tree (for comparison the inner most label for a regular MPLS Label Stack is the outgoing label). If the Receiving MPLS Label Stack sub-TLVe has more than one labels, the second inner most label in the stack identifies the expected upstream neighbor and explicit RPF checking needs to be set up for the tree label accordingly.

2.1.5. Tree Label sub-TLV

The MPLS Label Stack sub-TLV can be used to specify the complete label stack used to send to a downstream node, with the stack including both a transport label (stack) and a label that identifies the tree to the downstream node. There are cases where the controller only wants to specify the tree-identifying label but leave

the transport details to the router itself. For example, the router could locally determine a transport label (stack) and combine with the tree-identifying label signaled from the controller to get the complete outgoing label stack.

For that purpose, a new Tree Label sub-TLV is defined, with a one-octet length field. The value field contains a single label entry with the same format and usage of one label entry in the MPLS Label Stack sub-TLV.

2.2. Context Label Wide Community

For a router to signal the context label that it assigns for a controller (or any label allocator that assigns labels that will be seen by this router), it attaches a Context Label Wide Community [I-D.ietf-idr-wide-bgp-communities] to the host route for its own address used in its BGP session towards the controllers (directly or via RRs). This is a new wide community that specifies the (Label Allocator, Context Label) tuple, and the exact format will be specified in a future revision.

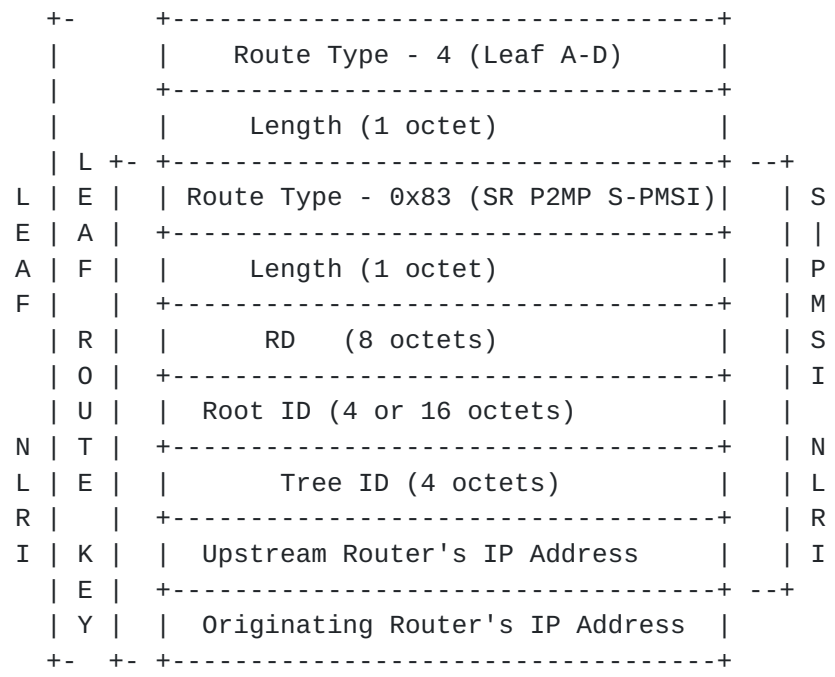
2.3. SR P2MP Signaling

An SR P2MP policy for an SR P2MP tree is identified by a (Root, Tree-id) tuple. It has a set of leaves and set of Candidate Paths (CPs). The policy is instantiated on the root of the tree, with corresponding Replication Segments - identified by (Root, Tree-id, Tree-Node-id) - instantiated on the tree nodes (root, leaves, and intermediate replication points). The Candidate Path is implicitly identified by the Route Distinguisher.

2.3.1. S-PMSI A-D Route for SR P2MP

With BGP signaled IP multicast trees and mLDP tunnels, the tree/tunnel identification is encoded in the NLRI of S-PMSI A-D routes and corresponding Leaf A-D routes. The signaling sets up forwarding state on each node of the tree, so the NLRI also contains the identisfication of the node in the "Upstream Router's IP Address" field.

For SR P2MP, forwarding state are represented as Replication Segments and are signaled from controllers to tree nodes. A Replication Segment is identified in a new type of S-PMSI A-D route and corresponding Leaf A-D route (note that the "Leaf" term here does not refer to tree leaves):

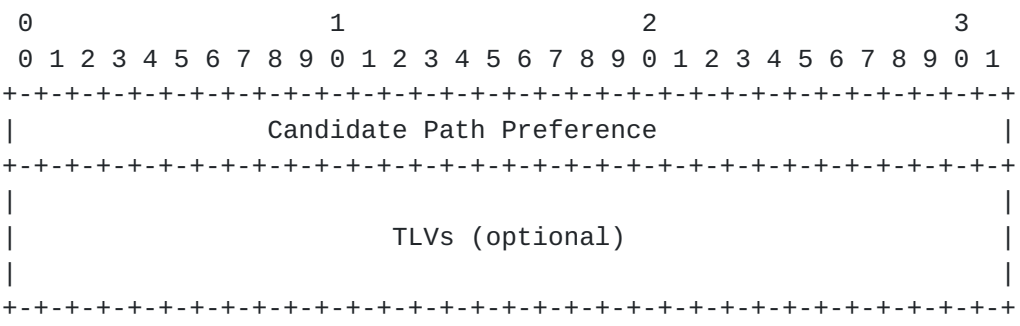


Leaf A-D route for SR Replication Segment

2.3.2. BGP Community Container for SR P2MP Policy

The Leaf A-D route for Replication Segments signaled to the root is also used to signal (parts of) the SR P2MP Policy - the policy name, the set of leaves (optional, for informational purpose), preference of the CP and other information are all encoded in a newly defined BGP Community Container (BCC) [I-D.ietf-idr-wide-bgp-communities] called SR P2MP Policy BCC.

The SR P2MP Policy BCC has a BGP Community Container type to be assigned by IANA. It is composed of a fixed 4-octet Candidate Path Preference value, optionally followed by TLVs.



BGP Community Container for SR P2MP Policy

One optional TLV is to enclose the following optional Atoms TLVs that are already defined in [I-D.ietf-idr-wide-bgp-communities]:

- An IPv4 or IPv6 Prefix list - for the set of leaves
- A UTF-8 string - for the policy name

If more information for the policy are needed, more Atoms TLVs or SR P2MP Policy BCC specific TLVs can be defined.

The root receives one Leaf A-D route for each Candidate Path of the policy. Only one of the routes need to, though more than one MAY include the above listed optional Atom TLVs in the SR P2MP Policy BCC.

2.3.3. SR Policy Tunnel Type

The Tunnel Encapsulation Attribute (TEA) attached to Leaf A-D routes encodes all replication branch information. For example, if an SR explicit path is to be used to reach a particular downstream node, the TEA will include a tunnel that lists the entire label stack for that SR path, plus the label that identifies the SR P2MP tree to the downstream node.

That SR path may have been installed on the node as a unicast SR policy with a corresponding Binding SID. In stead of listing the entire label stack in an MPLS tunnel in the TEA, a different tunnel, SR Policy Tunnel [I-D.ietf-idr-segment-routing-te-policy], can be used as an alternative. The tunnel includes a Binding SID sub-TLV, an optional endpoint sub-TLV that identifies the downstream node, and an optional one-segment segment list that identifies to the downstream node the SR P2MP tree. When a node receives the Leaf A-D route with the TEA that contains an SR Policy Tunnel without a RPF sub-TLV, the Binding SID is used to locate corresponding outgoing segment lists used to reach the downstream node; the tree-identifying

segment from the optional one-segment segment list is added to to outgoing segment lists mapped from the binding SID to form the entire segment list used to send traffic to downstream node.

Note that, the SR Policy Tunnel is initially defined to instantiate an SR policy. For that use case it provides information associated with the policy, e.g., Binding SID, preference, and segment lists. The receiving node installs that policy and establishes the mapping from the Binding SID to the outgoing segments. The use of SR Policy Tunnel in this document is to refer to a pre-installed SR policy so the preference and segment lists are not used.

If a tunnel in the TEA carries a RPF sub-TLV, it is for the upstream node. The tunnel may be an MPLS tunnel in case of SR MPLS, and the Receiving MPLS Label Stack sub-TLV specifies the incoming label stack that identifies the tree and optionally the upstream neighbor. Alternatively, for both SR-MPLS and SRv6 an SR Policy Tunnel with the RPF sub-TLV can be used, in which the Binding SID sub-TLV is the SID for the tree.

If the node is the root and a Binding SID is allocated by the controller, the Binding SID is signaled to the root in a TEA tunnel with a RPF sub-TLV as above but without a destination sub-TLV.

3. Procedures

Details to be added. The general idea is described in the introduction section.

4. Security Considerations

This document does not introduce new security risks.

5. IANA Considerations

This document makes the following IANA requests:

- Assign "Any-Encapsulation" and "Load-balancing" tunnel types from the "BGP Tunnel Encapsulation Attribute Tunnel Types" registry

- Assign "Member Tunnels", "Receiving MPLS Label Stack", "Tree label" and "RPF" sub-TLV types from the "BGP Tunnel Encapsulation Attribute Sub-TLVs" registry. The "Member Tunnels" sub-TLV has a two-octet value length (so the type should be in the 128-255 range), while the "Receiving MPLS Label Stack", "Tree Label" and "RPF" sub-TLV has a one-octet value length.

- Assign "S-PMSI A-D Route for SR P2MP" route type from the "BGP MCAST-TREE Route Types" registry, with a suggested value of 0x83.

- Assign a new BGP Community Container type "SR P2MP Policy", and to create an "SR P2MP Policy Community Container TLV Registry", with an initial entry for "TLV for Atoms".

6. Acknowledgements

The authors Eric Rosen for his questions, suggestions, and help finding solutions to some issues like the neighbor based explicit RPF checking. The authors also thank Lenny Giuliano, Sanoj Vivekanandan and IJsbrand Wijnands for their review and comments.

7. References

7.1. Normative References

[I-D.ietf-bess-bgp-multicast]
Zhang, Z., Giuliano, L., Patel, K., Wijnands, I., mishra, m., and A. Gulko, "BGP Based Multicast", draft-ietf-bess-bgp-multicast-02 (work in progress), June 2020.

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

[I-D.ietf-idr-tunnel-encaps]
Patel, K., Velde, G., Sangli, S., and J. Scudder, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-17 (work in progress), July 2020.

[I-D.ietf-idr-wide-bgp-communities]
Raszuk, R., Haas, J., Lange, A., Decraene, B., Amante, S., and P. Jakma, "BGP Community Container Attribute", draft-ietf-idr-wide-bgp-communities-05 (work in progress), July 2018.

[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.

[I-D.voyer-spring-sr-replication-segment]
Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "SR Replication Segment for Multi-point Service Delivery", draft-voyer-spring-sr-replication-segment-04 (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>.

[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

[RFC6388] Wijnands, IJ., Ed., Minei, I., Ed., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, DOI 10.17487/RFC6388, November 2011, <https://www.rfc-editor.org/info/rfc6388>.

[RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <https://www.rfc-editor.org/info/rfc6513>.

[RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <https://www.rfc-editor.org/info/rfc7761>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

Authors' Addresses

Zhaohui Zhang
Juniper Networks

EMail: zzhang@juniper.net

Robert Raszuk
Bloomberg LP

EMail: robert@raszuk.net

Dante Pacella
Verizon

EMail: dante.j.pacella@verizon.com

Arkadiy Gulko
Refinitiv

EMail: arkadiy.gulko@refinitiv.com