Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 29, 2015

J. Dong
M. Chen
Huawei Technologies
R. Raszuk
Mirantis Inc.

November 25, 2014

Extensions to RT-Constrain in Hierarchical Route Reflection Scenarios
draft-dong-idr-rtc-hierarchical-rr-02

Abstract

The Route Target (RT) Constrain mechanism specified in RFC 4684 is used to build a route distribution graph in order to restrict the propagation of Virtual Private Network (VPN) routes. In network scenarios where hierarchical route reflection (RR) is used, the existing RT-Constrain mechanism cannot build a correct route distribution graph. This document describes the problem scenario and proposes a solution to address the RT-Constrain issue in hierarchical RR scenarios.

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 is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

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

This Internet-Draft will expire on May 29, 2015.

Table of Contents

1. Introduction

The Route Target (RT) Constrain mechanism specified in [RFC4684] is used to build a route distribution graph in order to restrict the propagation of Virtual Private Network (VPN) routes. In network scenarios where hierarchical route reflection (RR) is used, the existing advertisment rules of RT membership information as defined in section 3.2 of [RFC4684] cannot guarantee a correct route distribution graph.

This document describes the problem scenario and proposes a solution to address the RT-Constrain issue in hierarchical RR scenarios.

2. Problem Statement

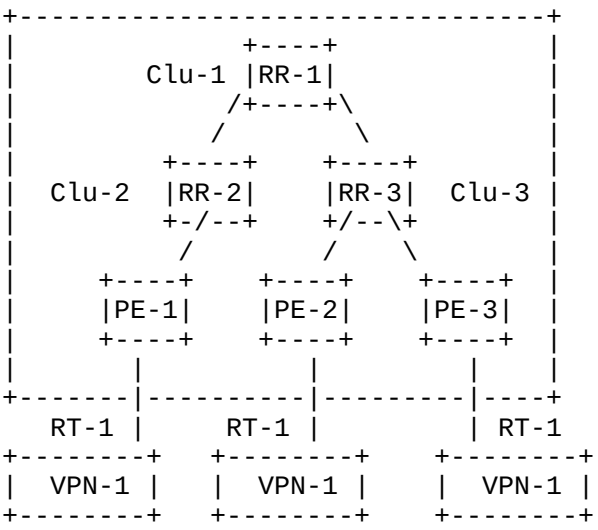


Figure 1. RT-Constrain with Hierarchical RR

As shown in Figure 1, hierarchical RRs are deployed in the network, RR-2 and RR-3 are route-reflectors of their connecting PEs, and are also the clients of RR-1. If each PE advertises RT membership information of RT-1 to the upstream RR, after the best path selection, both RR-2 and RR-3 would create the CLUSTER_LIST attribute, prepend their local CLUSTER_ID and then advertise the best path to RR-1 and their clients respectively.

On receipt of the RT-Constrain routes from RR-2 and RR-3, RR-1 will select one of the received routes as the best route, here assume the route received from RR-2 is selected by RR-1 as the best path. Then RR-1 needs to advertise the best path to both RR-2 and RR-3 to create the route distribution graph of VPN-1. RR-1 would prepend its CLUSTER_ID to the CLUSTER_LIST of the path, and according to the rules in Section 3.2 of [RFC4684], it sets the ORIGINATOR_ID to its own router-id, and the NEXT_HOP to the local address for the session. Then RR-1 would advertise this route to both RR-2 and RR-3. On receipt of the RT-Constrain route from RR-1, RR-2 checks the CLUSTER_LIST and find its own CLUSTER_ID in the list, so this route will be ignored by RR-2. As a result, RR-2 will not form the outbound filter of RT-1 towards RR-1, hence will not advertise VPN routes with RT-1 to RR-1.

3. Proposed Solution

3.1. Add-path Based Solution

During the discussion in the IDR working group, the add-path based solution is proposed. It makes use of the add-path mechanism as defined in [I-D.ietf-idr-add-paths] for RTC route advertisement. The solution is summerized as follows:

- The route-reflector clients which themselves are also route-reflectors SHOULD be identified, then BGP add-paths [I-D.ietf-idr-add-paths] SHOULD be enabled for RT membership NLRI on the BGP sessions between the higher layer RR and the lower layer RRs to ensure that sufficient RT-Constrain routes can be advertised by the higher layer RR to the lower layer RRs to pass BGP loop detection. In this case normal BGP path advertisement rules as defined in [RFC4271] SHOULD be applied. The number of RT-Constrain routes to be advertised is a local decision of operators.

- When advertising an RT membership NLRI to a route-reflector client which is not a lower layer RR, the advertisement rule as defined in section 3.2 of [RFC4684] SHOULD be applied.

With the above advertisement rule, RR-1 in figure 1 SHOULD advertise to RR-2 the RT-Constrain routes received from both RR-2 and RR-3, then the RTC route from RR-3 will pass the BGP loop detection on RR-2, thus the route distribution graph can be set up correctly.

4. IANA Considerations

This document makes no request of IANA.

5. Security Considerations

This document does not change the security properties of BGP based VPNs and [RFC4684].

6. Acknowledgements

The authors would like to thank Yaqun Xiao for the discussion about RT-Constrain in hierarchical RR scenario. Many people have made valuable comments and suggestions, including Susan Hares, Jeffrey Haas, Stephane Litkowski, Vitkovsky Adam, Xiaohu Xu, Uttaro James, Shyam Sethuram and Saikat Ray.

7. Normative References

[I-D.ietf-idr-add-paths]
Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", draft-ietf-idr-add-paths-10 (work in progress), October 2014.

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

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[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, November 2006.

Appendix A. Another Possible Solution

This section provides another possible solution which was discussed among authors and IDR participants.

Since the advertisement of RT-Constrain route is to set up a route distribution graph and not to guide the data packet forwarding, actually all the available RT-Constrain routes should be considered in setting up the route distribution graph, not just the best one. Thus the following advertisment rule for RT membership information is proposed to replace the rule i and ii in section 3.2 [RFC4684]:

- When advertising an RT membership NLRI to a route-reflector peer (either client or non-client), if the best path as selected by the path selection procedure described in Section 9.1 of [RFC4271] is the path received from this peer, and there are alternative paths received from other peers, then the most disjoint alternative route SHOULD be advertised to this peer. The most disjoint alternative path is the path whose CLUSTER_LIST and ORIGINATOR_ID attributes are diverse from the attributes of the best path.

With the above advertisement rule, RR-1 in figure 1 would advertise to RR-2 the RT-Constrain route received from RR-3, although the best route is received from RR-2. Thus RR-2 will not discard the RT-constrain route received from RR-1, and the route distribution graph can be set up correctly.

Authors' Addresses

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

Email: jie.dong@huawei.com

Mach(Guoyi) Chen
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: mach.chen@huawei.com

Robert Raszuk
Mirantis Inc.
615 National Ave. #100
Mt View, CA 94043
USA

Email: robert@raszuk.net

Inter-Domain Routing
Internet-Draft
Intended status: Informational
Expires: April 9, 2018

H. Gredler, Ed.
RtBrick Inc.
K. Vairavakkalai
C. Ramachandran
B. Rajagopalan
E. Aries
Juniper Networks, Inc.
L. Fang
eBay

October 06, 2017

Egress Peer Engineering using BGP-LU
draft-gredler-idr-bgplu-epe-11

Abstract

The MPLS source routing paradigm provides path control for both intra- and inter- Autonomous System (AS) traffic. RSVP-TE is utilized for intra-AS path control. This documents outlines how MPLS routers may use the BGP labeled unicast protocol (BGP-LU) for doing traffic-engineering on inter-AS links.

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 is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

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

This Internet-Draft will expire on April 9, 2018.

Table of Contents

1. Introduction

Today, BGP-LU [RFC3107] is used both as an intra-AS [I-D.ietf-mpls-seamless-mpls] and inter-AS routing protocol. BGP-LU may advertise a MPLS transport path between IGP regions and Autonomous Systems. Those paths may span one or more router hops. This document describes advertisement and use of one-hop MPLS label-switched paths (LSPs) for traffic-engineering the links between Autonomous Systems.

Consider Figure 1: an ASBR router (R2) advertises a labeled host route for the remote-end IP address of its link (IP3). The BGP next-hop gets set to R2s loopback IP address. For the advertised Label <N> a forwarding action of 'POP and forward' to next-hop (IP3) is installed in R2's MPLS forwarding table. Now consider if R2 had several links and R2 would advertise labels for all of its inter-AS links. By pushing the corresponding MPLS label <N> on the label-stack an ingress router R1 may control the egress peer selection.

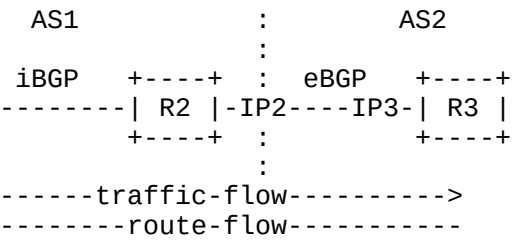


Figure 1: single-hop LSPs

Of course, since R1 and R2 may not be directly connected to each other, if the interior routers within AS1 do not maintain routes to external destinations, carrying traffic to such destinations would require a tunnel from R1 to R2. Such tunnel could be realized as either a MPLS Label Switched Path (LSP), or by GRE [RFC2784].

2. Motivation, Rationale and Applicability

BGP-LU is often just seen as a 'stitching' protocol for connecting Autonomous Systems. BGP-LU is often not viewed as a viable protocol for solving the Inter-domain traffic-engineering problem.

With this document the authors want to clarify the use of BGP-LU for Egress Peering traffic-engineering purposes and encourage both implementers and network operators to use a widely deployed and operationally well understood protocol, rather than inventing new protocols or new extensions to the existing protocols.

3. Sample Topology

The following topology (Figure 2) and IP addresses shall be used throughout the Egress Peering Engineering advertisement examples.

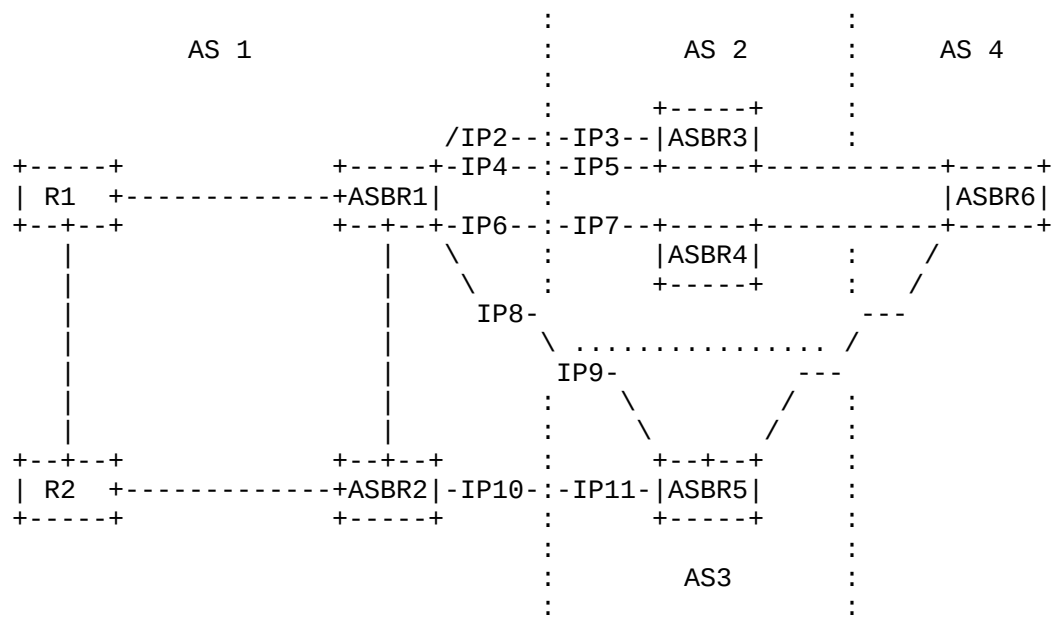


Figure 2: Sample Topology

3.1. Loopback IP addresses and Router-IDs

- R1: 192.0.2.1/32
- R2: 192.0.2.2/32
- ASBR1: 192.0.2.11/32
- ASBR2: 192.0.2.12/32
- ASBR3: 192.0.2.13/32
- ASBR4: 192.0.2.14/32
- ASBR5: 192.0.2.15/32
- ASBR6: 192.0.2.16/32

3.2. Link IP addresses

- ASBR1 (203.0.113.2/31) to ASBR3 (203.0.113.3/31) link #1
- ASBR1 (203.0.113.4/31) to ASBR3 (203.0.113.5/31) link #2
- ASBR1 (203.0.113.6/31) to ASBR4 (203.0.113.7/31)
- ASBR1 (203.0.113.8/31) to ASBR5 (203.0.113.9/31)
- ASBR2 (203.0.113.10/31) to ASBR5 (203.0.113.11/31)

4. Service Route Advertisement

In Figure 3 a simple network layout is shown. There are two classes of BGP speakers:

1. Ingress Routers
2. Controllers

Ingress routers receive BGP-LU routes from the ASBRs. Each BGP-LU route corresponds to an egress link. Furthermore Ingress routers receive their service routes using the BGP protocol. The BGP Add-paths extension [I-D.ietf-idr-add-paths] ensures that multiple paths to a given service route may get advertised.

As outlined in [I-D.filsfils-spring-segment-routing-central-epe], Controllers receive BGP-LU routes from the ASBRs as well. However the service routes may be received either using the BGP protocol plus the BGP Add-paths extension [I-D.ietf-idr-add-paths] or alternatively The BGP Monitoring protocol [I-D.ietf-grow-bmp] (BMP). BMP has support for advertising the RIB-In of a BGP router. As such it might be a suitable protocol for feeding all potential egress paths of a service-route from a ASBR into a controller.

5. Egress Next-hop Advertisement

An ASBR assigns a distinct label for each of its next-hops facing an eBGP peer and advertises it to its internal BGP mesh. The ASBR programs a forwarding action ’POP and forward’ into the MPLS forwarding table. Note that the neighboring AS is not required to support exchanging NLRIs with the local AS using BGP-LU. It is the local ASBR (ASBR{1,2}) which generates the BGP-LU routes into its iBGP mesh or controller facing session(s). The forwarding next-hop for those routes points to the link-IP addresses of the remote ASBRs (ASBR{3,4,5}). Note that the generated BGP-LU routes always match the BGP next-hop that the remote ASBRs set their BGP service routes

to, such that the software component doing route-resolution understands the association between the BGP service route and the BGP-LU forwarding route.

5.1. iBGP meshing and BGP nexthop rewrite policy

Throughout this document we describe how the BGP next-hop of both BGP Service Routes and BGP-LU routes shall be rewritten. This may clash with existing network deployments and existing network configurations guidelines which may mandate to rewrite the BGP next-hop when an BGP update enters an AS.

The Egress peering use case assumes a central controller as shown Figure 3. In order to support both existing BGP nexthop guidelines and the suggestion described in this document, an implementation SHOULD support several internal BGP peer-groups:

1. iBGP peer group for Ingress Routers

2. iBGP peer group for Controllers

The first peer group MAY be left unchanged and use any existing BGP nexthop rewrite policy. The second peer group MUST use the BGP rewrite policy described in this document for both service and BGP-LU routes.

Of course a common iBGP peer group and a common rewrite policy may be used if the proposed policy is compatible with existing routing software implementations of BGP next-hop route resolution.

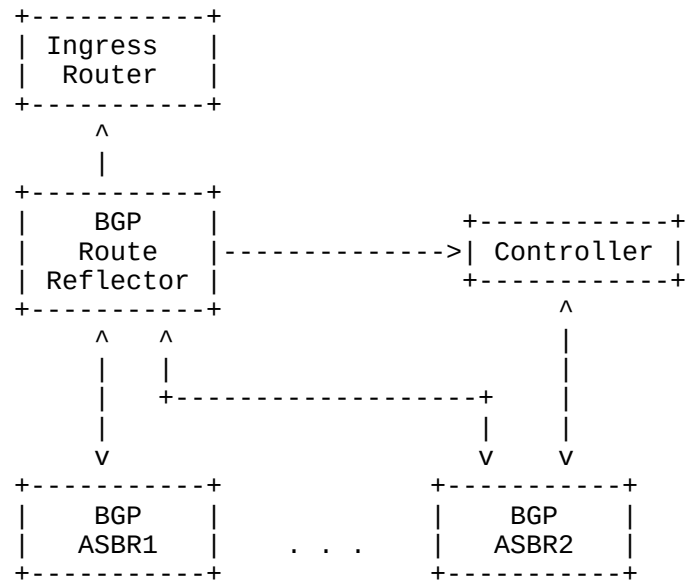


Figure 3: Selective iBGP NH rewrite

5.2. Single-hop eBGP

In Figure 2 the ASBR{1,5} and ASBR{2,5} links are examples for single-hop eBGP advertisements.

- ASBR5 advertises a BGP service (SAFI-1) route {172.16/12} to ASBR1 with a BGP next-hop of 203.0.113.9. When ASBR1 re-advertises this BGP service route towards its iBGP mesh (R{1,2}) it does not overwrite the BGP next-hop, but rather leaves it unchanged.

- ASBR1 advertises a BGP-LU route {203.0.113.9/32, label 100} with a BGP next hop of 192.0.2.11. ASBR1 programs a MPLS forwarding state of 'POP and forward' to 203.0.113.9 for the advertised label 100.

- ASBR5 advertises a BGP service (SAFI-1) route {172.16/12} to ASBR2 with a BGP next-hop of 203.0.113.11. When ASBR2 re-advertises this BGP service route towards its iBGP mesh (R{1,2}) it does not overwrite the BGP next-hop, but rather leaves it unchanged.

- ASBR2 advertises BGP-LU route {203.0.113.11/32, label 101} with a BGP next hop of 192.0.2.12. ASBR2 programs a MPLS forwarding state of 'POP and forward' to 203.0.113.11 for the advertised label 101.

- Should the operator already be redistributing egress links into the network for purposes of BGP next-hop resolution, the BGP-LU route {203.0.113.9/32, label 100} will now take precedence due to LPM over the previous redistributed prefix {203.0.113.8/31}. If the BGP next-hop prefix {203.0.113.9/32} were to be redistributed as-is, then standard protocol best-path and preference selection mechanisms will be exhausted in order to select the best-path.

- In general, ASBR1 may receive advertisements for the route to 172.16/12 from ASBR3 and ASBR4, as well as from ASBR5. One of these other advertisements may be chosen as the best path by the BGP decision process. In order to allow ASBR1 to re-advertise the route to 172.16/12 received from ASBR5 with next-hop 203.0.113.9, independent of the other advertisements received, ASBR1 and R{1,2} need to support the BGP add-paths extension. [I-D.ietf-idr-add-paths].

5.3. Multi-hop eBGP

Todays operational practice for load-sharing across parallel links is to configure a single multi-hop eBGP session between a pair of routers. The IP addresses for the Multi-hop eBGP session are typically sourced from the loopback IP interfaces. Note that those IP addresses do not share an IP subnet. Most often those loopback IP addresses are most specific host routes. Since the BGP next-hops of the received BGP service routes are typically rewritten to the remote routers loopback IP address they cannot get immediatly resolved by the receiving router. To overcome this, the operator configures a static route with next-hops pointing to each of the remote-IP addresses of the underlying links.

In Figure 2 both ASBR{1,3} links are examples of a multi-hop eBGP advertisement. In order to advertise a distinct label for a common FEC throughout the iBGP mesh, ASBR1 and all the receiving iBGP routers need to support the BGP Add-paths extension. [I-D.ietf-idr-add-paths].

- ASBR3 advertises a BGP service (SAFI-1) route {172.16/12} over multi-hop eBGP to ASBR1 with a BGP next-hop of 192.0.2.13. When ASBR1 re-advertises this BGP service route towards its iBGP mesh (R{1,2}) it does not overwrite the BGP next-hop, but rather leaves it unchanged. Note that the iBGP routers SHOULD support the BGP Add-paths extension [I-D.ietf-idr-add-paths] such that ASBR can re-advertise all paths to the SAFI-1 route {172.16/12}.

- For link #1, ASBR1 advertises into its iBGP mesh a BGP-LU route {192.0.2.13/32, label 102} with a BGP next hop of 192.0.2.11. To differentiate this from the link #2 route-advertisement (which

contains the same FEC) it is setting the path-ID to 1. ASBR1 programs a MPLS forwarding state of 'POP and forward' to 203.0.113.3 for the advertised label 102.

- For link #2, ASBR1 advertises into its iBGP mesh a BGP-LU route {192.0.2.13/32, label 103} with a BGP next hop of 192.0.2.11. To differentiate this from the link #1 route-advertisement (which contains the same FEC) it is setting the path-ID to 2. ASBR1 programs a MPLS forwarding state of 'POP and forward' to 203.0.113.5 for the advertised label 103.

- Should the operator already be redistributing static routes into the network, the BGP next-hop {192.0.2.13} may already be resolvable. It is then that standard protocol best-path and preference selection mechanisms will be exhausted in order to select the best-path.

5.4. Grouping of Peers

In addition to offering a distinct BGP-LU label for each egress link, an ASBR MAY want to advertise a BGP-LU label which represents a load-balancing forwarding action across a set of peers. The difference is here that the ingress node gives up individual link control, but rather delegates the load-balancing decision to a particular egress router which has the freedom to send the traffic down to any link in the Peer Set as identified by the BGP-LU label.

Assume that ASBR1 wants to advertise a label identifying the Peer Set {ASBR3, ASBR4, ASBR5}.

- For the two ASBR{1,3} links in Figure 2, belonging to Peer Set 1, ASBR1 advertises a single BGP-LU route {192.0.2.13/32, label 104} with a BGP next hop of 192.0.2.11. To differentiate this from the ASBR{1,3} single link route-advertisements (which contains the same FEC) it is setting the path-ID to 3 and attaching a Peer-Set Community 'Peer Set 1'.

- For the ASBR{1,4} link in Figure 2, ASBR1 advertises a BGP-LU route {203.0.113.7/32, label 104} with a BGP next hop of 192.0.2.11. To differentiate this from the ASBR{1,4} single link route-advertisements (which contains the same FEC) it is setting the path-ID to 2 and attaching a Peer-Set Community 'Peer Set 1'.

- For the ASBR{1,5} link in Figure 2, ASBR1 advertises a BGP-LU route {203.0.113.9/32, label 104} with a BGP next hop of 192.0.2.11. To differentiate this from the ASBR{1,5} single link route-advertisements (which contains the same FEC) it is setting the path-ID to 2 and attaching a Peer-Set Community 'Peer Set 1'.

Finally ASBR1 programs a MPLS forwarding state of 'POP and load-balance' to {203.0.113.3, 203.0.113.5, 203.0.113.7, 203.0.113.9} for the advertised label 104.

5.5. Supporting Summarization at ASBR

5.5.1. Locality forwarding bias

A router has one or more forwarding plane units. A forwarding plane unit consists of one or more interfaces. Forwarding of packets to an interface that is member of a forwarding plane unit is cheaper than across units.

A route entry in the forwarding-table may contain multiple next-hops, each pointing to a network-interface. When forwarding a packet, a forwarding plane unit may optionally provide preference to a subset of these next-hops, whose interfaces are its own members. This behavior is called "Locality forwarding bias".

5.5.2. Label per group of peers sharing a locality

An ASBR MAY assign a distinct label for the set of eBGP-peers that share a forwarding plane unit and advertise it to its internal BGP mesh. The ASBR programs a forwarding action 'POP and IP-lookup' into the MPLS forwarding table for these labels. While performing the IP-lookup, the ASBR MUST perform "Locality-forwarding bias" to ensure it only selects next-hops towards eBGP peers that are attached to the current forwarding plane unit, where the IP-lookup is happening.

This provides the ingress-peers with ability to steer traffic towards a "subset of eBGP-peers" attached to an ASBR, while preserving the ability of the ASBR to aggregate the IP prefixes received from those eBGP-peers, while re-advertising to the internal BGP mesh.

6. Egress Link Protection

It is desirable to provide a local-repair based protection scheme, in case a redundant path is available to reach a peer AS. Protection may be applied at multiple levels in the routing stack. Since the ASBR has insight into both BGP-LU and BGP service advertisements, protection can be provided at the BGP-LU, at the BGP service or both levels.

6.1. FRR backup routes

Assume the network operator wants to provide a local-repair next-hop for the 172.16/12 BGP service route at ASBR1. The active route resolves over the parallel links towards ASBR3. In case the link #1

between ASBR{1,3} fails there are now several candidate backup paths providing protection against link or node failure.

6.1.1. Local links

Assuming that the remaining link #2 between ASBR{1,3} has enough capacity, and link-protection is sufficient, this link MAY serve as temporary backup.

However if node-protection or additional capacity is desired, then the local link between ASBR{1,4} or ASBR{1,5} MAY be used as temporary backup.

6.1.2. Remote BGP-LU labels

ASBR1 is both originator and receiver of BGP routing information. For this protection method it is required that the ASBRs support the [I-D.ietf-idr-best-external] behavior. ASBR1 receives both the BGP-LU and BGP service routes from ASBR2 and therefore can use the ASBR2 advertised label as a backup path given that ASBR1 has a tunnel towards ASBR2.

6.1.3. Local IP forwarding tables

For protecting plain unicast (Internet) routing information a very simple backup scheme could be to recurse to the relevant IP forwarding table and do an IP lookup to further determine a new egress link.

6.2. Avoiding micro-loops during FRR

Typically, Egress Link Protection mechanisms for Service-routes at the ASBRs are susceptible to micro forwarding-loops if the IP-lookup at backup-path ASBR points back to the primary-ASBR for some reason, during local-repair period.

By using mechanisms described in this document, such forwarding-loops can be avoided. Because the backup-ASBR will receive a MPLS-packet with EPE label, it will not do an IP-lookup, and will forwarding traffic based on MPLS-label lookup only. Thus the repaired traffic is guaranteed to exit the network towards an Egress-peer at backup-ASBR, and not turn back towards the IBGP core.

7. Dynamic link utilization

For a software component which controls the egress link selection it may be desirable to know about a particular egress links current

utilization, such that it can adjust the traffic that gets sent to a particular interface.

In [I-D.ietf-idr-link-bandwidth] a community for reporting link-bandwidth is specified. Rather than reporting the static bandwidth of the link, the ASBRs shall report the available bandwidth as seen by the data-plane via the link-bandwidth community in their BGP-LU update message.

It is crucial that ingress routers learn quickly about congestion of an egress link and hence it is desired to get timely updates of the advertised per-link BGP-LU routes carrying the available bandwidth information when the available bandwidth crosses a certain (preconfigured) threshold.

Controllers may also utilize the link-bandwidth community among other common mechanisms to retrieve data-plane statistics (e.g. SNMP, NETCONF)

8. Acknowledgements

Many thanks to Yakov Rekhter, Chris Bowers and Jeffrey (Zhaohui) Zhang for their detailed review and insightful comments.

Special thanks to Richard Steenbergen and Tom Scholl who brought up the original idea of using MPLS for BGP based egress load-balancing at their inspiring talk at Nanog 48.

9. IANA Considerations

This documents does not request any action from IANA.

10. Security Considerations

This document does not introduce any change in terms of BGP security.

11. References

11.1. 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>.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <https://www.rfc-editor.org/info/rfc2784>.

[RFC3107] Rekhter, Y. and E. Rosen, "Carrying Label Information in BGP-4", RFC 3107, DOI 10.17487/RFC3107, May 2001, <https://www.rfc-editor.org/info/rfc3107>.

11.2. Informative References

[I-D.filsfils-spring-segment-routing-central-epe] Filsfils, C., Previdi, S., Patel, K., Shaw, S., Ginsburg, D., and D. Afanasiev, "Segment Routing Centralized Egress Peer Engineering", draft-filsfils-spring-segment-routing-central-epe-05 (work in progress), August 2015.

[I-D.ietf-grow-bmp] Scudder, J., Fernando, R., and S. Stuart, "BGP Monitoring Protocol", draft-ietf-grow-bmp-17 (work in progress), January 2016.

[I-D.ietf-idr-add-paths] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", draft-ietf-idr-add-paths-15 (work in progress), May 2016.

[I-D.ietf-idr-best-external] Marques, P., Fernando, R., Chen, E., Mohapatra, P., and H. Gredler, "Advertisement of the best external route in BGP", draft-ietf-idr-best-external-05 (work in progress), January 2012.

[I-D.ietf-idr-link-bandwidth] Mohapatra, P. and R. Fernando, "BGP Link Bandwidth Extended Community", draft-ietf-idr-link-bandwidth-06 (work in progress), January 2013.

[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.

Authors' Addresses

Hannes Gredler (editor)
RtBrick Inc.

Email: hannes@rtbrick.com

Kaliraj Vairavakkalai
Juniper Networks, Inc.
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: kaliraj@juniper.net

Chandra Ramachandran
Juniper Networks, Inc.
Electra, Exora Business Park Marathahalli - Sarjapur Outer Ring Road
Bangalore, KA 560103
India

Email: csekar@juniper.net

Balaji Rajagopalan
Juniper Networks, Inc.
Electra, Exora Business Park Marathahalli - Sarjapur Outer Ring Road
Bangalore, KA 560103
India

Email: balajir@juniper.net

Ebben Aries
Juniper Networks, Inc.
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: earies@juniper.net

Luyuan Fang
eBay
411 108th Ave NE
Bellevue, WA 98004
US

Email: lufang@ebay.com

IDR
Internet-Draft
Intended status: Standards Track
Expires: January 21, 2016

K. Patel
S. Previdi
C. Filsfils
A. Sreekantiah
Cisco Systems
S. Ray
Unaffiliated
H. Gredler
Juniper Networks
July 20, 2015

Segment Routing Prefix SID extensions for BGP
draft-keyupate-idr-bgp-prefix-sid-05

Abstract

Segment Routing (SR) architecture allows a node to steer a packet flow through any topological path and service chain by leveraging source routing. The ingress node prepends a SR header to a packet containing a set of "segments". Each segment represents a topological or a service-based instruction. Per-flow state is maintained only at the ingress node of the SR domain.

This document describes the BGP extension for announcing BGP Prefix Segment Identifier (BGP Prefix SID) information.

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] only when they appear in all upper case. They may also appear in lower or mixed case as English words, without any normative meaning.

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 21, 2016.

Table of Contents

1. Segment Routing Documents

The main references for this document are the SR architecture defined in [I-D.ietf-spring-segment-routing] and the related use case illustrated in [I-D.filsfils-spring-segment-routing-msdc].

The Segment Routing Egress Peer Engineering architecture is described in [I-D.filsfils-spring-segment-routing-central-epe].

The Segment Routing Egress Peer Engineering BGPLS extensions are described in [I-D.ietf-idr-bgpls-segment-routing-epe].

2. Introduction

Segment Routing (SR) architecture leverages the source routing paradigm. A group of inter-connected nodes that use SR forms a SR domain. The ingress node of the SR domain prepends a SR header containing "segments" to an incoming packet. Each segment represents a topological instruction such as "go to prefix P following shortest path" or a service instruction (e.g.: "pass through deep packet inspection"). By inserting the desired sequence of instructions, the ingress node is able to steer a packet via any topological path and/or service chain; per-flow state is maintained only at the ingress node of the SR domain.

Each segment is identified by a Segment Identifier (SID). As described in [I-D.ietf-spring-segment-routing], when SR is applied to the MPLS dataplane the SID consists of a label while when SR is applied to the IPv6 dataplane the SID consists of an IPv6 prefix (see [I-D.previdi-6man-segment-routing-header]).

A BGP-Prefix Segment (aka BGP-Prefix-SID), is a BGP segment attached to a BGP prefix. A BGP-Prefix-SID is always global within the SR/BGP domain and identifies an instruction to forward the packet over the ECMP-aware best-path computed by BGP to the related prefix. The BGP-Prefix-SID is the identifier of the BGP prefix segment.

This document describes the BGP extension to signal the BGP-Prefix-SID. Specifically, this document defines a new BGP attribute known as the BGP Prefix SID attribute and specifies the rules to originate, receive and handle error conditions of the new attribute.

As described in [I-D.filsfils-spring-segment-routing-msdc], the newly proposed BGP Prefix-SID attribute can be attached to prefixes from AFI/SAFI:

Multiprotocol BGP labeled IPv4/IPv6 Unicast ([RFC3107]).

Multiprotocol BGP ([RFC4760]) unlabeled IPv6 Unicast.

[I-D.filsfils-spring-segment-routing-msdc] describes use cases where the Prefix-SID is used for the above AFI/SAFI.

3. BGP-Prefix-SID

The BGP-Prefix-SID attached to a BGP prefix P represents the instruction "go to Prefix P" along its BGP bestpath (potentially ECMP-enabled).

3.1. MPLS Prefix Segment

The BGP Prefix Segment is realized on the MPLS dataplane in the following way:

According to [I-D.ietf-spring-segment-routing], each BGP speaker is configured with a label block called the Segment Routing Global Block (SRGB). The SRGB of a node is a local property and could be different on different speakers.

As described in [I-D.filsfils-spring-segment-routing-msdc] the operator assigns a globally unique "index", L_I, to a locally sourced prefix of a BGP speaker N which is advertised to all other BGP speakers in the SR domain.

The index L_I is a 32 bit offset in the SRGB. Each BGP speaker derives its local MPLS label, L, by adding L_I to the start value of its own SRGB, and programs L in its MPLS dataplane as its incoming/local label for the prefix.

If the BGP speakers are configured with the same SRGB start value, they will all program the same MPLS label for a given prefix P. This has the effect of having a single label for prefix P across all BGP speakers despite that the MPLS paradigm of "local label" is preserved and this clearly simplifies the deployment and operations of traffic engineering in BGP driven networks, as described in [I-D.filsfils-spring-segment-routing-msdc].

If the BGP speakers cannot be configured with the same SRGB, the proposed BGP Prefix-SID attribute allows the advertisement of the SRGB so each node can advertise the SRGB it’s configured with. The drawbacks of the use case where BGP speakers have different SRGBs are documented in [I-D.filsfils-spring-segment-routing-msdc].

In order to advertise the label index of a given prefix P and, optionally, the SRGB, a new extension to BGP is needed: the BGP

Prefix SID attribute. This extension is described in subsequent sections.

3.2. IPv6 Prefix Segment

As defined in [I-D.previdi-6man-segment-routing-header], in SR for the IPv6 dataplane, the SRGB consists of the set of IPv6 addresses used within the SR domain (as described in [I-D.previdi-6man-segment-routing-header]). Therefore the BGP speaker willing to process SR IPv6 packets MUST advertise an IPv6 prefix with the attached Prefix SID attribute and related SR IPv6 flag (see subsequent section).

As described in [I-D.filsfils-spring-segment-routing-msdc], when SR is used over an IPv6 dataplane, the BGP Prefix Segment is instantiated by an IPv6 prefix originated by the BGP speaker.

Each node advertises a globally unique IPv6 address representing itself in the domain. This prefix (e.g.: its loopback interface address) is advertised to all other BGP speakers in the SR domain.

Also, each node MUST advertise its support of Segment Routing for IPv6 dataplane. This is realized using the Prefix SID Attribute defined below.

4. BGP-Prefix-SID Attribute

The BGP Prefix SID attribute is an optional, transitive BGP path attribute. The attribute type code is to be assigned by IANA (suggested value: 40). The value field of the BGP-Prefix-SID attribute has the following format:

The value field of the BGP Prefix SID attribute is defined here to be a set of elements encoded as "Type/Length/Value" (i.e., a set of TLVs). Following TLVs are defined:

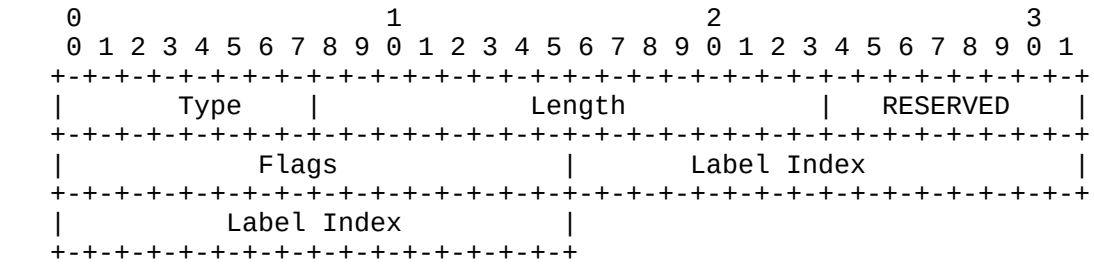
- Label-Index TLV
- IPv6 SID TLV
- Originator SRGB TLV

Label-Index and Originator SRGB TLVs are used only when SR is applied to the MPLS dataplane.

IPv6 SID TLV is used only when SR is applied to the IPv6 dataplane.

4.1. Label-Index TLV

The Label-Index TLV MUST be present in the Prefix-SID attribute attached to Labeled IPv4/IPv6 unicast prefixes ([RFC3107]) and has the following format:

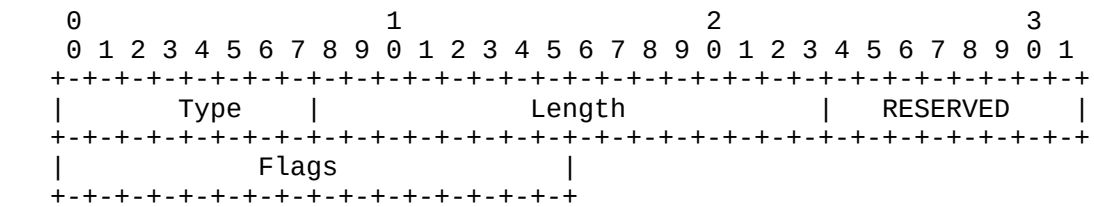


where:

- Type is 1.
- Length: is 7, the total length of the value portion of the TLV.
- RESERVED: 8 bit field. SHOULD be 0 on transmission and MUST be ignored on reception.
- Flags: 16 bits of flags. None are defined at this stage of the document. The flag field SHOULD be clear on transmission and MUST be ignored at reception.
- Label Index: 32 bit value representing the index value in the SRGB space.

4.2. IPv6 SID

The Label-Index TLV MUST be present in the Prefix-SID attribute attached to MP-BGP unlabeled IPv6 unicast prefixes ([RFC4760]) and has the following format:



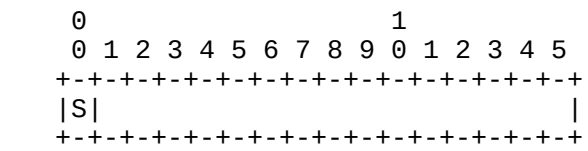
where:

- Type is 2.

- Length: is 3, the total length of the value portion of the TLV.

- RESERVED: 8 bit field. SHOULD be 0 on transmission and MUST be ignored on reception.

- Flags: 16 bits of flags defined as follow:



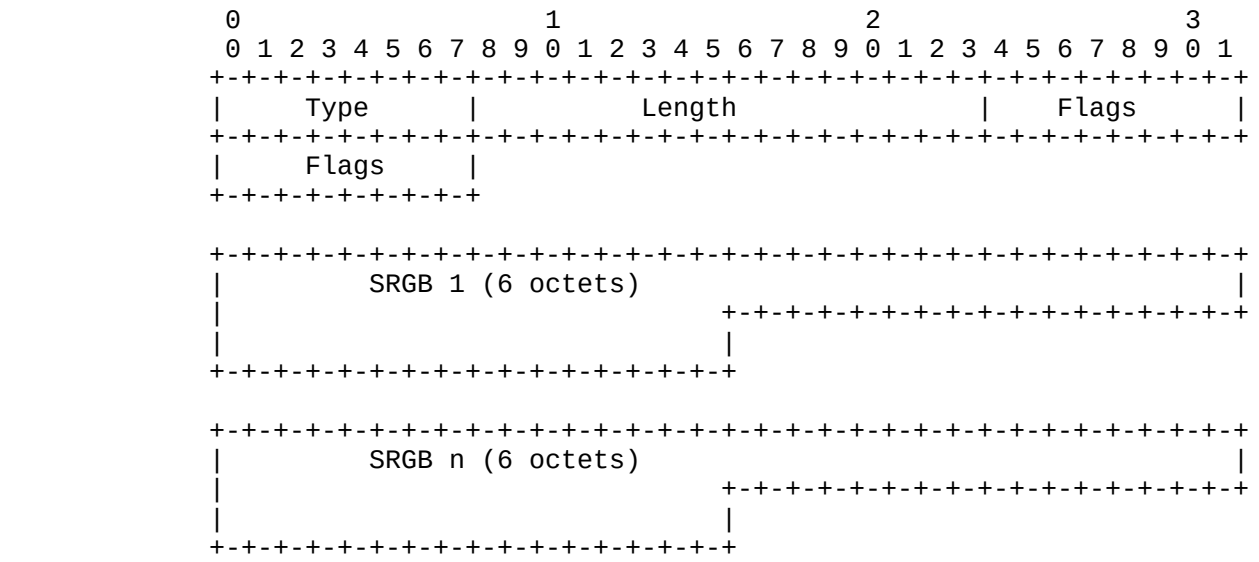
where:

  - S flag: if set then it means that the BGP speaker attaching the Prefix-SID Attribute to a prefix is capable of processing the IPv6 Segment Routing Header (SRH, [I-D.previdi-6man-segment-routing-header]) for the segment corresponding to the originated IPv6 prefix. The use case leveraging the S flag is described in [I-D.filsfils-spring-segment-routing-msdc].

The other bits of the flag field SHOULD be clear on transmission an MUST be ignored at reception.

4.3. Originator SRGB TLV

The Originator SRGB TLV is an optional TLV and has the following format:



where:

- Type is 3.
- Length is the total length of the value portion of the TLV: 2 + multiple of 6.
- Flags: 16 bits of flags. None are defined in this document. Flags SHOULD be clear on transmission an MUST be ignored at reception.
- SRGB: 3 octets of base followed by 3 octets of range. Note that SRGB field MAY appear multiple times.

The Originator SRGB TLV contains the SRGB of the router originating the prefix to which the BGP Prefix SID is attached and MUST be kept in the Prefix-SID Attribute unchanged during the propagation of the BGP update.

The originator SRGB describes the SRGB of the node where the BGP Prefix Segment end. It is used to build SRTE policies when different SRGB's are used in the fabric ([I-D.filsfils-spring-segment-routing-msdc]).

The originator SRGB may only appear on Prefix-SID attribute attached to prefixes of SAFI 4 (labeled unicast, [RFC3107]).

5. Receiving BGP-Prefix-SID Attribute

A BGP speaker receiving a BGP Prefix-SID attribute from an EBGP neighbor residing outside the boundaries of the SR domain, SHOULD discard the attribute unless it is configured to accept the attribute from the EBGP neighbor. A BGP speaker MAY log an error for further analysis when discarding an attribute.

5.1. MPLS Dataplane: Labeled Unicast

The Prefix-SID attribute MUST contain the Label-Index TLV and MAY contain the Originator SRGB. A BGP Prefix-SID attribute received without a Label-Index TLV MUST be considered as "unacceptable" by the receiving speaker.

A BGP speaker may be locally configured with an SRGB=[GB_S, GB_E]. The preferred method for deriving the SRGB is a matter of local router configuration.

Given a label index L_I, we call L = L_I + GB_S as the derived label. A BGP Prefix-SID attribute is called "unacceptable" for a speaker M if the derived label value L lies outside the SRGB configured on M. Otherwise the Label Index attribute is called "acceptable" to speaker M.

The mechanisms through which a given label_index value is assigned to a given prefix are outside the scope of this document. The label-index value associated with a prefix is locally configured at the BGP router originating the prefix.

The Prefix-SID attribute MUST contain the Label-Index TLV and MAY contain the Originator SRGB TLV. A BGP Prefix-SID attribute received without a Label-Index TLV MUST be considered as "unacceptable" by the receiving speaker.

When a BGP speaker receives a path from a neighbor with an acceptable BGP Prefix-SID attribute, it SHOULD program the derived label as the local label for the prefix in its MPLS dataplane. In case of any error, a BGP speaker MUST resort to the error handling rules specified in Section 7. A BGP speaker MAY log an error for further analysis.

When a BGP speaker receives a path from a neighbor with an unacceptable BGP Prefix-SID attribute, for the purpose of label allocation, it SHOULD treat the path as if it came without a Prefix-SID attribute. A BGP speaker MAY choose to assign a local (also called dynamic) label (non-SRGB) for such a prefix. A BGP speaker MAY log an error for further analysis.

A BGP speaker receiving a prefix with a Prefix-SID attribute and a label NLRI field of implicit-null from a neighbor MUST adhere to standard behavior and program its MPLS dataplane to pop the top label when forwarding traffic to the prefix. The label NLRI defines the outbound label that MUST be used by the receiving node. The Label Index gives a hint to the receiving node on which local/incoming label the BGP speaker SHOULD use.

5.2. IPv6 Dataplane

When a SR IPv6 BGP speaker receives a IPv6 Unicast BGP Update with a prefix having the BGP Prefix SID attribute attached, it checks whether the IPv6 SID TLV is present and if the S-flag is set. If the IPv6 SID TLV is not present or if the S-flag is not set, then the Prefix-SID attribute MUST be considered as "unacceptable" by the receiving speaker.

The Originator SRGB MUST be ignored on reception.

A BGP speaker receiving a BGP Prefix-SID attribute from an EBGP neighbor residing outside the boundaries of the SR domain, SHOULD discard the attribute unless it is configured to accept the attribute from the EBGP neighbor. A BGP speaker MAY log an error for further analysis when discarding an attribute.

6. Announcing BGP-Prefix-SID Attribute

The BGP Prefix-SID attribute MAY be attached to labeled BGP prefixes (IPv4/IPv6) [RFC3107]or to IPv6 prefixes [RFC4760]. In order to prevent distribution of the BGP Prefix-SID attribute beyond its intended scope of applicability, attribute filtering MAY be deployed.

6.1. MPLS Dataplane: Labeled Unicast

A BGP speaker that originates a prefix attaches the Prefix-SID attribute when it advertises the prefix to its neighbors. The value of the Label-Index in the Label-Index TLV is determined by configuration.

A BGP speaker that originates a Prefix-SID attribute MAY optionally announce Originator SRGB TLV along with the mandatory Label-Index TLV. The content of the Originator SRGB TLV is determined by the configuration.

Since the Label-index value must be unique within an SR domain, by default an implementation SHOULD NOT advertise the BGP Prefix-SID attribute outside an Autonomous System unless it is explicitly configured to do so.

A BGP speaker that advertises a path received from one of its neighbors SHOULD advertise the Prefix-SID received with the path without modification regardless of whether the Prefix-SID was acceptable. If the path did not come with a Prefix-SID attribute, the speaker MAY attach a Prefix-SID to the path if configured to do so. The content of the TLVs present in the Prefix-SID is determined by the configuration.

In all cases, the label field of the NLRI ([RFC3107], [RFC4364]) MUST be set to the local/incoming label programmed in the MPLS dataplane for the given prefix. If the prefix is associated with one of the BGP speakers interfaces, this label is the usual MPLS label (such as the implicit or explicit NULL label).

6.2. IPv6 Dataplane

A BGP speaker that originates a prefix attaches the Prefix-SID attribute when it advertises the prefix to its neighbors. The IPv6 SID TLV MUST be present and the S-flag MUST be set.

A BGP speaker that advertises a path received from one of its neighbors SHOULD advertise the Prefix-SID received with the path without modification regardless of whether the Prefix-SID was acceptable. If the path did not come with a Prefix-SID attribute, the speaker MAY attach a Prefix-SID to the path if configured to do so. The IPv6-SID TLV MUST be present in the Prefix-SID and with the S-flag set.

7. Error Handling of BGP-Prefix-SID Attribute

When a BGP Speaker receives a BGP Update message containing a malformed BGP Prefix-SID attribute, it MUST ignore the received BGP Prefix-SID attributes and not pass it to other BGP peers. This is equivalent to the -attribute discard- action specified in [I-D.ietf-idr-error-handling]. When discarding an attribute, a BGP speaker MAY log an error for further analysis.

If the BGP Prefix-SID attribute appears more than once in an BGP Update message message, then, according to [I-D.ietf-idr-error-handling], all the occurrences of the attribute other than the first one SHALL be discarded and the BGP Update message shall continue to be processed.

When a BGP speaker receives an unacceptable Prefix-SID attribute, it MAY log an error for further analysis.

8. IANA Considerations

This document defines a new BGP path attribute known as the BGP Prefix-SID attribute. This document requests IANA to assign a new attribute code type (suggested value: 40) for BGP the Prefix-SID attribute from the BGP Path Attributes registry.

This document defines two new TLVs for BGP Prefix-SID attribute. These TLVs need to be registered with IANA. We request IANA to create a new registry for BGP Prefix-SID Attribute TLVs as follows:

Under "Border Gateway Protocol (BGP) Parameters" registry, "BGP Prefix SID attribute Types" Reference: draft-keyupate-idr-bgp-prefix-side-05 Registration Procedure(s): Values 1-254 First Come, First Served, Value 0 and 255 reserved

Value	Type	Reference
0	Reserved	draft-keyupate-idr-bgp-prefix-side-05
1	Label-Index	draft-keyupate-idr-bgp-prefix-side-05
2	IPv6 SID	draft-keyupate-idr-bgp-prefix-side-05
3	Originator SRGB	draft-keyupate-idr-bgp-prefix-side-05
4-254	Unassigned	
255	Reserved	draft-keyupate-idr-bgp-prefix-side-05

9. Security Considerations

This document introduces no new security considerations above and beyond those already specified in [RFC4271] and [RFC3107].

10. Acknowledgements

The authors would like to thanks Satya Mohanty and Acee Lindem for their contribution to this document.

11. Change Log

Initial Version: Sep 21 2014

12. References

12.1. Normative References

[I-D.ietf-idr-error-handling]
Chen, E., Scudder, J., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP UPDATE Messages", draft-ietf-idr-error-handling-19 (work in progress), April 2015.

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

[RFC3107] Rekhter, Y. and E. Rosen, "Carrying Label Information in BGP-4", RFC 3107, DOI 10.17487/RFC3107, May 2001, <http://www.rfc-editor.org/info/rfc3107>.

[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, <http://www.rfc-editor.org/info/rfc4271>.

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

12.2. Informative References

[I-D.filsfils-spring-segment-routing-central-epe] Filsfils, C., Previdi, S., Patel, K., Aries, E., shaw@fb.com, s., Ginsburg, D., and D. Afanasiev, "Segment Routing Centralized Egress Peer Engineering", draft-filsfils-spring-segment-routing-central-epe-04 (work in progress), July 2015.

[I-D.filsfils-spring-segment-routing-msdc] Filsfils, C., Previdi, S., Mitchell, J., Aries, E., Lapukhov, P., Gaya, G., Afanasiev, D., Laberge, T., Nkposong, E., Nanduri, M., Uttaro, J., and S. Ray, "BGP-Prefix Segment in large-scale data centers", draft-filsfils-spring-segment-routing-msdc-02 (work in progress), July 2015.

[I-D.ietf-idr-bgpls-segment-routing-epe] Previdi, S., Filsfils, C., Ray, S., Patel, K., Dong, J., and M. Chen, "Segment Routing Egress Peer Engineering BGP-LS Extensions", draft-ietf-idr-bgpls-segment-routing-epe-00 (work in progress), June 2015.

[I-D.ietf-spring-segment-routing] Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-03 (work in progress), May 2015.

[I-D.previdi-6man-segment-routing-header]
Previdi, S., Filsfils, C., Field, B., and I. Leung, "IPv6 Segment Routing Header (SRH)", draft-previdi-6man-segment-routing-header-06 (work in progress), May 2015.

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

Authors' Addresses

Keyur Patel
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95124 95134
USA

Email: keyupate@cisco.com

Stefano Previdi
Cisco Systems
Via Del Serafico, 200
Rome 00142
Italy

Email: sprevidi@cisco.com

Clarence Filsfils
Cisco Systems
Brussels
Belgium

Email: cfilsfils@cisco.com

Arjun Sreekantiah
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95124 95134
USA

Email: asreekan@cisco.com

Saikat Ray
Unaffiliated

Email: raysaikat@gmail.com

Hannes Gredler
Juniper Networks

Email: hannes@juniper.net

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 29, 2015

S. Previdi, Ed.
C. Filsfils
Cisco Systems, Inc.
S. Ray
Individual Contributor
K. Patel
Cisco Systems, Inc.
J. Dong
M. Chen
Huawei Technologies
April 27, 2015

Segment Routing Egress Peer Engineering BGP-LS Extensions
draft-previdi-idr-bgpls-segment-routing-epe-03

Abstract

Segment Routing (SR) leverages source routing. A node steers a packet through a controlled set of instructions, called segments, by prepending the packet with an SR header. A segment can represent any instruction, topological or service-based. SR allows to enforce a flow through any topological path and service chain while maintaining per-flow state only at the ingress node of the SR domain.

The Segment Routing architecture can be directly applied to the MPLS dataplane with no change on the forwarding plane. It requires minor extension to the existing link-state routing protocols.

This document outline a BGP-LS extension for exporting BGP egress point topology information (including its peers, interfaces and peering ASs) in a way that is exploitable in order to compute efficient Egress Point Engineering policies and strategies.

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 is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute

working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on October 29, 2015.

Table of Contents

1. Introduction

Segment Routing (SR) leverages source routing. A node steers a packet through a controlled set of instructions, called segments, by prepending the packet with an SR header. A segment can represent any instruction, topological or service-based. SR allows to enforce a flow through any topological path and service chain while maintaining per-flow state only at the ingress node of the SR domain.

The Segment Routing architecture can be directly applied to the MPLS dataplane with no change on the forwarding plane. It requires minor extension to the existing link-state routing protocols.

This document outline a BGP-LS extension for exporting BGP egress point topology information (including its peers, interfaces and peering ASs) in a way that is exploitable in order to compute efficient Egress Point Engineering policies and strategies.

This document defines new types of segments: a Peer Node segment describing the BGP session between two nodes; a Peer Adjacency Segment describing the link (one or more) that is used by the BGP session; the Peer Set Segment describing an arbitrary set of sessions or links between the local BGP node and its peers.

2. Segment Routing Documents

The main reference for this document is the SR architecture defined in [I-D.ietf-spring-segment-routing].

The Segment Routing Egress Peer Engineering architecture is described in [I-D.filsfils-spring-segment-routing-central-epe].

3. BGP Peering Segments

As defined in [draft-filsfils-spring-segment-routing-epe], an EPE enabled Egress PE node MAY advertise segments corresponding to its attached peers. These segments are called BGP peering segments or BGP Peering SIDs. They enable the expression of source-routed inter-domain paths.

An ingress border router of an AS may compose a list of segments to steer a flow along a selected path within the AS, towards a selected egress border router C of the AS and through a specific peer. At minimum, a BGP Peering Engineering policy applied at an ingress PE involves two segments: the Node SID of the chosen egress PE and then the BGP Peering Segment for the chosen egress PE peer or peering interface.

This document defines the BGP EPE Peering Segments: Peer Node, Peer Adjacency and Peer Set.

Each BGP session MUST be described by a Peer Node Segment. The description of the BGP session MAY be augmented by additional Adjacency Segments. Finally, each Peer Node Segment and Peer Adjacency Segment MAY be part of the same group/set so to be able to group EPE resources under a common Peer-Set Segment Identifier (SID).

Therefore, when the extensions defined in this document are applied to the use case defined in [I-D.filsfils-spring-segment-routing-central-epe]:

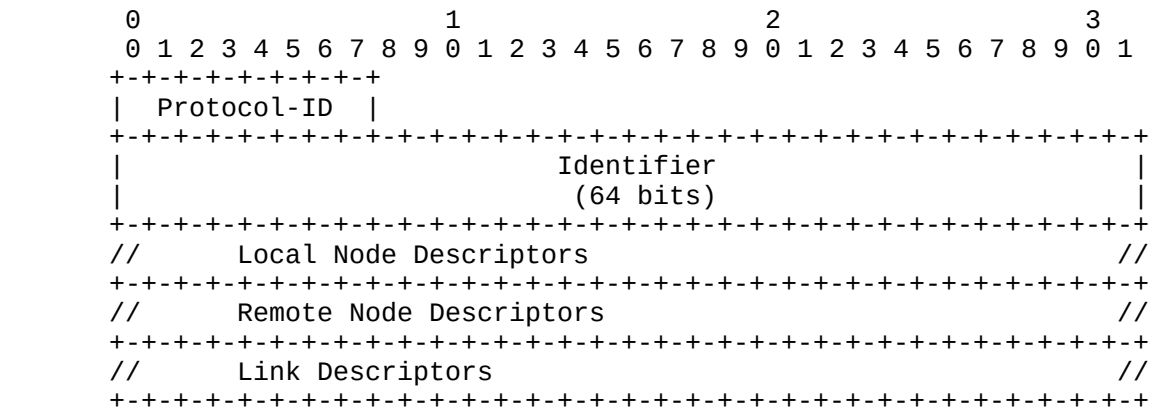
- One Peer Node Segment MUST be present.
- One or more Peer Adjacency Segments MAY be present.
- Each of the Peer Node and Peer Adjacency Segment MAY use the same Peer-Set.

4. Link NLRI for EPE Connectivity Description

This section describes the NLRI used for describing the connectivity of the BGP Egress router. The connectivity is based on links and remote peers/ASs and therefore the existing Link-Type NLRI (defined in [I-D.ietf-idr-ls-distribution]) is used. A new Protocol ID is used (codepoint to be assigned by IANA, suggested value 7).

The use of a new Protocol-ID allows separation and differentiation between the NLRIs carrying BGP-EPE descriptors from the NLRIs carrying IGP link-state information as defined in[I-D.ietf-idr-ls-distribution]. The Link NLRI Type uses descriptors and attributes already defined in [I-D.ietf-idr-ls-distribution] in addition to new TLVs defined in the following sections of this document.

The format of the Link NLRI Type is as follows:



Node Descriptors and Link Descriptors are defined in [I-D.ietf-idr-ls-distribution].

4.1. BGP Router ID and Member ASN

Two new Node Descriptors Sub-TLVs are defined in this document:

- BGP Router Identifier (BGP Router-ID):

  Type: TBA (suggested value 516).

  Length: 4 octets

  Value: 4 octet unsigned integer representing the BGP Identifier as defined in [RFC4271] and [RFC6286].

- Confederation Member ASN (Member-ASN)

  Type: TBA (suggested value 517).

  Length: 4 octets

  Value: 4 octet unsigned integer representing the Member ASN inside the Confederation.[RFC5065].

4.2. EPE Node Descriptors

The following Node Descriptors Sub-TLVs MUST appear in the Link NLRI as Local Node Descriptors:

- BGP Router ID, which contains the BGP Identifier of the local BGP EPE node.

- Autonomous System Number, which contains the local ASN or local confederation identifier (ASN) if confederations are used.

- BGP-LS Identifier.

The following Node Descriptors Sub-TLVs MAY appear in the Link NLRI as Local Node Descriptors:

- Member-ASN, which contains the ASN of the confederation member (when BGP confederations are used).

- Node Descriptors as defined in [I-D.ietf-idr-ls-distribution].

The following Node Descriptors Sub-TLVs MUST appear in the Link NLRI as Remote Node Descriptors:

- BGP Router ID, which contains the BGP Identifier of the peer node.

- Autonomous System Number, which contains the peer ASN or the peer confederation identifier (ASN), if confederations are used.

The following Node Descriptors Sub-TLVs MAY appear in the Link NLRI as Remote Node Descriptors:

- Member-ASN, which contains the ASN of the confederation member (when BGP confederations are used).

- Node Descriptors as defined in defined in [I-D.ietf-idr-ls-distribution].

4.3. Link Attributes

The following BGP-LS Link attributes TLVs are used with the Link NLRI:

TLV Code Point	Description	Length
1099	Adjacency-Segment Identifier (Adj-SID)	variable
TBA	Peer-Segment Identifier (Peer-SID)	variable
TBA	Peer-Set-SID	variable

Adj-SID is defined in [I-D.gredler-idr-bgp-ls-segment-routing-extension] and the same format is used for the Peer-SID and Peer-Set-SID TLVs.

Peer-SID and Peer-Set SID are two new sub-TLVs with the same format as the Adj-SID and whose codepoints are to be assigned by IANA:

Peer-SID: SID representing the peer of the BGP session. The format is the same as defined for the Adj-SID in [I-D.gredler-idr-bgp-ls-segment-routing-extension]. Suggested codepoint value: 1036

Peer-Set-SID: the SID representing the group the peer is part of. The format is the same as defined for the Adj-SID in [I-D.gredler-idr-bgp-ls-segment-routing-extension]. Suggested codepoint value: 1037

The value of the Adj-SID, Peer-SID and Peer-Set-SID Sub-TLVs SHOULD be persistent across router restart.

The Peer-SID MUST be present when BGP-LS is used for the use case described in [I-D.filsfils-spring-segment-routing-central-epe] and MAY be omitted for other use cases.

The Adj-SID and Peer-Set-SID SubTLVs MAY be present when BGP-LS is used for the use case described in [I-D.filsfils-spring-segment-routing-central-epe] and MAY be omitted for other use cases.

In addition, BGP-LS Nodes and Link Attributes, as defined in [I-D.ietf-idr-ls-distribution]MAY be inserted in order to advertise the characteristics of the link.

5. Peer Node and Peer Adjacency Segments

In this section the following Peer Segments are defined:

Peer Node Segment (Peer Node SID)

Peer Adjacency Segment (Peer Adj SID)

Peer Set Segment (Peer Set SID)

5.1. Peer Node Segment

The Peer Node Segment describes the BGP session peer (neighbor). It MUST be present when describing an EPE topology as defined in [I-D.filsfils-spring-segment-routing-central-epe]. The Peer Node

Segment is encoded within the BGP-LS Link NLRI specified in Section 4.

The Peer Node Segment is a local segment. At the BGP node advertising it, its semantic is:

- SR header operation: NEXT (as defined in [I-D.ietf-spring-segment-routing]).
- Next-Hop: the connected peering node to which the segment is related.

The Peer Node Segment is advertised with a Link NLRI, where:

- Local Node Descriptors contains

  Local BGP Router ID of the EPE enabled egress PE.
  Local ASN.
  BGP-LS Identifier.

- Remote Node Descriptors contains

  Peer BGP Router ID (i.e.: the peer BGP ID used in the BGP session).
  Peer ASN.

- Link Descriptors Sub-TLVs, as defined in [I-D.ietf-idr-ls-distribution], contain the addresses used by the BGP session:
  - IPv4 Interface Address (Sub-TLV 259) contains the BGP session IPv4 local address.
  - IPv4 Neighbor Address (Sub-TLV 260) contains the BGP session IPv4 peer address.
  - IPv6 Interface Address (Sub-TLV 261) contains the BGP session IPv6 local address.
  - IPv6 Neighbor Address (Sub-TLV 262) contains the BGP session IPv6 peer address.
- Link Attribute contains the Peer-SID TLV as defined in Section 4.3.
- In addition, BGP-LS Link Attributes, as defined in [I-D.ietf-idr-ls-distribution], MAY be inserted in order to advertise the characteristics of the link.

5.2. Peer Adjacency Segment

The Peer Adjacency Segment is a local segment. At the BGP node advertising it, its semantic is:

- SR header operation: NEXT (as defined in [I-D.ietf-spring-segment-routing]).
- Next-Hop: the interface peer address.

The Peer Adjacency Segment is advertised with a Link NLRI, where:

- Local Node Descriptors contains

  Local BGP Router ID of the EPE enabled egress PE.
  Local ASN.
  BGP-LS Identifier.

- Remote Node Descriptors contains

  Peer BGP Router ID (i.e.: the peer BGP ID used in the BGP session).
  Peer ASN.

- Link Descriptors Sub-TLVs, as defined in [I-D.ietf-idr-ls-distribution], contain the addresses used by the BGP session:
  - IPv4 Interface Address (Sub-TLV 259) contains the IPv4 address of the local interface used by the BGP session.
  - IPv4 Neighbor Address (Sub-TLV 260) contains the IPv4 address of the peer interface used by the BGP session.
  - IPv6 Interface Address (Sub-TLV 261) contains the IPv6 address of the local interface used by the BGP session.
  - IPv6 Neighbor Address (Sub-TLV 262) contains the IPv6 address of the peer interface used by the BGP session.
- Link attribute used with the Peer Adjacency SID contains the Adj-SID TLV as defined in Section 4.3.

In addition, BGP-LS Link Attributes, as defined in [I-D.ietf-idr-ls-distribution], MAY be inserted in order to advertise the characteristics of the link.

5.3. Peer Set Segment

The Peer Set Segment is a local segment. At the BGP node advertising it, its semantic is:

- SR header operation: NEXT (as defined in [I-D.ietf-spring-segment-routing]).
- Next-Hop: load balance across any connected interface to any peer in the related set.

The Peer Set Segment is advertised within a Link NLRI (describing a Peer Node Segment or a Peer Adjacency segment) as a BGP-LS attribute.

The Peer Set Attribute contains the Peer-Set-SID TLV, defined in Section 4.3 identifying the set of which the Peer Node Segment or Peer Adjacency Segment is a member.

6. Illustration

6.1. Reference Diagram

The following reference diagram is used throughout this document. The solution is described for IPv4 with MPLS-based segments.

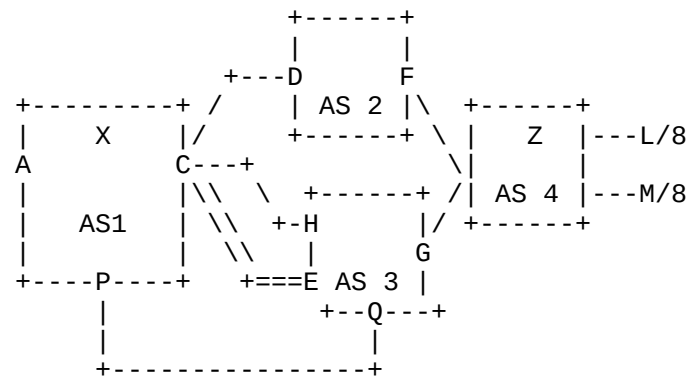


Figure 1: Reference Diagram

IPv4 addressing:

- C’s interface to D: 1.0.1.1/24, D’s interface: 1.0.1.2/24
- C’s interface to H: 1.0.2.1/24, H’s interface: 1.0.2.2/24
- C’s upper interface to E: 1.0.3.1/24, E’s interface: 1.0.3.2/24

- C’s lower interface to E: 1.0.4.1/24, E’s interface: 1.0.4.2/24
- Loopback of E used for eBGP multi-hop peering to C: 1.0.5.2/32
- C’s loopback is 3.3.3.3/32 with SID 64

BGP Router-IDs are C, D, H and E.

- C’s BGP Router-ID: 3.3.3.3
- D’s BGP Router-ID: 4.4.4.4
- E’s BGP Router-ID: 5.5.5.5
- H’s BGP Router-ID: 6.6.6.6

C’s BGP peering:

- Single-hop eBGP peering with neighbor 1.0.1.2 (D)
- Single-hop eBGP peering with neighbor 1.0.2.2 (H)
- Multi-hop eBGP peering with E on ip address 1.0.5.2 (E)

C’s resolution of the multi-hop eBGP session to E:

- Static route 1.0.5.2/32 via 1.0.3.2
- Static route 1.0.5.2/32 via 1.0.4.2

Node C configuration is such that:

- A Peer Node segment is allocated to each peer (D, H and E).
- An Adjacency segment is defined for each recursing interface to a multi-hop peer (CE upper and lower interfaces).
- A Peer Set segment is defined to include all peers in AS3 (peers H and E).

Local BGP-LS Identifier in router C is set to 10000.

The Link NLRI Type is used in order to encode C’s connectivity. the Link NLRI uses the new Protocol-ID value (to be assigned by IANA).

6.1.1. Peer Node Segment for Node D

Descriptors:

- Local Node Descriptors (BGP Router-ID, local ASN, BGP-LS Identifier): 3.3.3.3 , AS1, 10000
- Remote Node Descriptors (BGP Router-ID, peer ASN): 4.4.4.4, AS2
- Link Descriptors (BGP session IPv4 local address, BGP session IPv4 neighbor address): 1.0.1.1, 1.0.1.2

Attributes:

- Peer-SID: 1012
- Link Attributes: see section 3.3.2 of [I-D.ietf-idr-ls-distribution]

6.1.2. Peer Node Segment for Node H

Descriptors:

- Local Node Descriptors (BGP Router-ID, ASN, BGPL Identifier): 3.3.3.3 , AS1, 10000
- Remote Node Descriptors (BGP Router-ID ASN): 6.6.6.6, AS3
- Link Descriptors (BGP session IPv4 local address, BGP session IPv4 peer address): 1.0.2.1, 1.0.2.2

Attributes:

- Peer-SID: 1022
- Peer-Set-SID: 1060
- Link Attributes: see section 3.3.2 of [I-D.ietf-idr-ls-distribution]

6.1.3. Peer Node Segment for Node E

Descriptors:

- Local Node Descriptors (BGP Router-ID, ASN, BGP-LS Identifier): 3.3.3.3 , AS1, 10000
- Remote Node Descriptors (BGP Router-ID, ASN): 5.5.5.5, AS3

- Link Descriptors (BGP session IPv4 local address, BGP session IPv4 peer address): 3.3.3.3, 1.0.5.2

Attributes:

- Peer-SID: 1052
- Peer-Set-SID: 1060

6.1.4. Peer Adj Segment for Node E, Link 1

Descriptors:

- Local Node Descriptors (BGP Router-ID, ASN, BGP-LS Identifier): 3.3.3.3 , AS1, 10000
- Remote Node Descriptors (BGP Router-ID, ASN): 5.5.5.5, AS3
- Link Descriptors (IPv4 local interface address, IPv4 peer interface address): 1.0.3.1 , 1.0.3.2

Attributes:

- Adj-SID: 1032
- LinkAttributes: see section 3.3.2 of [I-D.ietf-idr-ls-distribution]

6.1.5. Peer Adj Segment for Node E, Link 2

Descriptors:

- Local Node Descriptors (BGP Router-ID, ASN, BGP-LS Identifier): 3.3.3.3 , AS1, 10000
- Remote Node Descriptors (BGP Router-ID, ASN): 5.5.5.5, AS3
- Link Descriptors (IPv4 local interface address, IPv4 peer interface address): 1.0.4.1 , 1.0.4.2

Attributes:

- Adj-SID: 1042
- LinkAttributes: see section 3.3.2 of [I-D.ietf-idr-ls-distribution]

7. BGP-LS EPE TLV/Sub-TLV Code Points Summary

The following table contains the TLVs/Sub-TLVs defined in this document.

Suggested Codepoint	Description	Defined in:
7	Protocol-ID	Section 4
516	BGP Router ID	Section 4.1
517	BGP Confederation Member	Section 4.1
1036	Peer-SID	Section 4.3
1037	Peer-Set-SID	Section 4.3

Table 1: Summary Table of BGP-LS EPE Codepoints

8. IANA Considerations

This document defines:

Two new Node Descriptors Sub-TLVs: BGP-Router-ID and BGP Confederation Member.

A new Protocol-ID for EPE: BGP-EPE.

Two new BGP-LS Attribute Sub-TLVs: the Peer-SID and the Peer-Set-SID.

The codepoints are to be assigned by IANA.

9. Manageability Considerations

TBD

10. Security Considerations

[I-D.ietf-idr-ls-distribution] defines BGP-LS NLRIs to which the extensions defined in this document apply.

The Security Section of [I-D.ietf-idr-ls-distribution] also applies to the:

new Node Descriptors Sub-TLVs: BGP-Router ID and BGP Confederation Member;

Peer-SID and Peer-Set-SID attributes

defined in this document.

11. Acknowledgements

The authors would like to thank Acee Lindem, Jakob Heitz, Howard Yang and Hannes Gredler for their feedback and comments.

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.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC5065] Traina, P., McPherson, D., and J. Scudder, "Autonomous System Confederations for BGP", RFC 5065, August 2007.

[RFC6286] Chen, E. and J. Yuan, "Autonomous-System-Wide Unique BGP Identifier for BGP-4", RFC 6286, June 2011.

12.2. Informative References

[I-D.filsfils-spring-segment-routing-central-epe] Filsfils, C., Previdi, S., Patel, K., Aries, E., shaw@fb.com, s., Ginsburg, D., and D. Afanasiev, "Segment Routing Centralized Egress Peer Engineering", draft-filsfils-spring-segment-routing-central-epe-03 (work in progress), January 2015.

[I-D.gredler-idr-bgp-ls-segment-routing-extension] Gredler, H., Ray, S., Previdi, S., Filsfils, C., Chen, M., and J. Tantsura, "BGP Link-State extensions for Segment Routing", draft-gredler-idr-bgp-ls-segment-routing-extension-02 (work in progress), October 2014.

[I-D.ietf-idr-ls-distribution] Gredler, H., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and TE Information using BGP", draft-ietf-idr-ls-distribution-10 (work in progress), January 2015.

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Shakir, R., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-ietf-spring-segment-routing-01 (work in progress), February 2015.

Authors' Addresses

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

Email: sprevidi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Brussels
BE

Email: cfilsfil@cisco.com

Saikat Ray
Individual Contributor

Email: raysaikat@gmail.com

Keyur Patel
Cisco Systems, Inc.
170, West Tasman Drive
San Jose, CA 95134
US

Email: keyupate@cisco.com

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

Email: jie.dong@huawei.com

Mach (Guoyi) Chen
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: mach.chen@huawei.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 14, 2015

A. Retana
Cisco Systems, Inc.

November 10, 2014

Advertisement of Multiple Paths in BGP: Implementation Report draft-retana-idr-add-paths-implementation-00

Abstract

This document reports the results of an ADD-PATH implementation survey. The survey had 22 questions about implementations' support for advertising multiple paths in BGP. After a brief summary of the results, each response is listed. This document contains responses from three implementers who completed the survey (Cumulus Networks, Cisco Systems and Exa Networks).

The editor did not use external means to verify the accuracy of the information submitted by the respondents. The respondents are considered experts on the products they reported on.

Status of This Memo

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

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

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

This Internet-Draft will expire on May 14, 2015.

Copyright Notice

Table of Contents

1. Introduction

This document reports results from a survey of implementations of the Advertisement of Multiple Paths in BGP [I-D.ietf-idr-add-paths], where a BGP [RFC4271] extension that allows the advertisement of multiple paths for the same address prefix without the new paths implicitly replacing any previous ones is defined. The essence of the extension is that each path is identified by a path identifier in addition to the address prefix.

The ADD-PATH implementation survey had 22 detailed questions about compliance with [I-D.ietf-idr-add-paths]. Three implementers (Cumulus Networks, Cisco Systems and Exa Networks) completed the survey. Section 4 provides a compilation of the results. Section 3.1 provides an overview of the differences between the implementations. Section 3.3 provides interoperability information.

2. Requirements Language

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

3. Results of the Survey

The respondents replied "Yes" or "No" to the survey’s questions to indicate whether their implementation supports the Functionality/ Description of the [RFC2119] language in [I-D.ietf-idr-add-paths]. The respondents replied "Other" to indicate an alternate behavior and had the opportunity to provide comments in all cases. Some questions were informative.

3.1. Overview of Differences

This section provides the reader with a shortcut to the points where the implementations differ.

The following questions were not answered "Yes" by all respondents (Note that the question numbers correspond to the subsection numbers of Section 4):

MUST

4.1.3, 4.1.4, 4.4.6

Question 4.1.3 asks about the ability of the implementation to uniquely identify a path. This question is linked to 4.1.2 in which the mechanism used to assigned Path Identifiers is explained. The

vendor that did not answer "Yes" to 4.1.3 lets the user assign Path Identifiers; the response to 4.1.3 was "Other: This is left to the user of the application to do."

Question 4.1.4 asks about the generation of Path Identifiers when re-advertising a route. All responded chose "Other" -- I believe that there was some misinterpretation on the intent of re-advertisement.

Question 4.4.6 asks about using the encoding defined when generating an Update. One vendor replied "Other"; in their case, transmitting Updates hasn't been implemented.

3.2. Implementation Identification

3.3.1. Cumulus

Company/Organization Name: Cumulus Networks

Implementation Name/Version: quagga

Date: 11/3/2014

Contact Name: Daniel Walton

Contact e-mail: dwalton@cumulusnetworks.com

3.3.2. Cisco

Company/Organization Name: Cisco Systems

Implementation Name/Version: IOS-XE

Date: 11/03/2014

Contact Name: Mohammed Mirza

Contact e-mail: mohamirz@cisco.com

3.3.3. Exa

Company/Organization Name: Exa Networks

Implementation Name/Version: ExaBGP

Date: 01/11/2014

Contact Name: Thomas Mangin

Contact e-mail: thomas.mangin@exa-networks.co.uk

3.3. Implementations and Interoperability

	Cumulus	Cisco	Exa	Other
Cumulus		Yes		Bird
Cisco		Yes		
Exa		Yes		

4. Implementation Report

For every item listed, the respondents indicated whether their implementation supports the Functionality/Description or not (Yes/No) according to the [RFC2119] language indicated. Any respondent comments are included. If appropriate, the respondents indicated with "Other" the fact that the support is neither Yes/No (an alternate behavior, for example). Refer to the appropriate sections in [I-D.ietf-idr-add-paths] for additional details.

4.1. Section 2: How to Identify a Path

4.1.1. Base Behavior

Functionality/Description: Is your implementation compatible with the use of the Path Identifier as described in this section?

[RFC2119]: N/A

Implementation	Yes/No/Other	Comments
Cumulus	Yes	
Cisco	Yes	
Exa	Yes	

4.1.2. Path Identifier Assignment

Functionality/Description: Explain how Path Identifiers are assigned in your implementation.

[RFC2119]: N/A

Implementation	Comments
Cumulus	quagga is RX only for now so this is not an issue
Cisco	Each net has unique path-id per paths under it. The path ids that are withdrawn can get assigned to the newer paths.
Exa	By the user

4.1.3. Path Identifier Assignment (2)

Functionality/Description: "...the Path Identifier MUST be assigned in such a way that the BGP speaker is able to use the (prefix, path identifier) to uniquely identify a path advertised to a neighbor."

Can your implementation uniquely identify an advertised path based on the (prefix, path identifier) pair?

[RFC2119]: MUST

Implementation	Yes/No/Other	Comments
Cumulus	Yes	
Cisco	Yes	
Exa	Other	This is left to the user of the application to do.

4.1.4. Route Re-advertisement

Functionality/Description: "A BGP speaker that re-advertises a route MUST generate its own Path Identifier to be associated with the re-advertised route."

Does your implementation generate a new Path Identifier when re-advertising a route?

[RFC2119]: MUST

Implementation	Yes/No/Other	Comments
Cumulus	Other	Comments quagga does not support TX yet
Cisco	Other	Once a BGP speaker advertises a path-id it has to also withdraw it. In case it has to readvertise, it either updates the older path-id path or creates a new path with a new unique id.
Exa	Other	ExaBGP does not re-advertise routes

4.1.5. Received Path Identifier

Functionality/Description: "A BGP speaker that receives a route SHOULD NOT assume that the identifier carries any particular semantics; it SHOULD be treated as an opaque value."

Does your implementation treat a received Path Identifier as an opaque value?

[RFC2119]: SHOULD NOT

Implementation	Yes/No/Other	Comments
Cumulus	Yes	
Cisco	Yes	
Exa	Yes	

4.2. Section 3: Extended NLRI Encodings

4.2.1. Base Behavior

Functionality/Description: Does your implementation use the encodings specified in this section?

[RFC2119]: N/A

Implementation	Yes/No/Other	Comments
Cumulus	Yes	
Cisco	Yes	
Exa	Yes	

4.3. Section 4: ADD-PATH Capability

4.3.1. Base Behavior

Functionality/Description: Is your implementation able to send and receive the ADD-PATH Capability as described in this section?

[RFC2119]: N/A

Implementation	Yes/No/Other	Comments
Cumulus	Yes	
Cisco	Yes	
Exa	Yes	

4.4. Section 5: Operation

4.4.1. Base Behavior

Functionality/Description: Is your implementation compatible with the operation described in this section?

[RFC2119]: N/A

Implementation	Yes/No/Other	Comments
Cumulus	Other	RX yes, TX not implemented
Cisco	Yes	
Exa	Yes	

4.4.2. Implicit Replacement

Functionality/Description: "...a new advertisement for a given address prefix and a given path identifier replaces a previous advertisement for the same address prefix and path identifier."

Does your implementation replace previous advertisements with the same (prefix, path identifier) pair?

[RFC2119]: N/A

Implementation	Yes/No/Other	Comments
Cumulus	Yes	
Cisco	Yes	
Exa	Other	ExaBGP does not implement a FIB

4.4.3. Silently Ignore

Functionality/Description: "If a BGP speaker receives a message to withdraw a prefix with a path identifier not seen before, it SHOULD silently ignore it."

Does your implementation silently ignore the withdraw of a prefix with a new path identifier?

[RFC2119]: SHOULD

Implementation	Yes/No/Other	Comments
Cumulus		
Cisco		
Exa		

4.4.4. Send/Receive Logic

Functionality/Description: "For a BGP speaker to be able to send multiple paths to its peer, that BGP speaker MUST advertise the ADD-PATH capability with the Send/Receive field set to either 2 or 3, and MUST receive from its peer the ADD-PATH capability with the Send/Receive field set to either 1 or 3, for the corresponding <AFI, SAFI>."

Does your implementation follow the send/receive logic as specified in this section?

[RFC2119]: MUST

Implementation	Yes/No/Other	Comments
Cumulus	Yes	
Cisco	Yes	
Exa	Yes	

4.4.5. Update Procedure

Functionality/Description: "A BGP speaker MUST follow the existing procedures in generating an UPDATE message for a particular <AFI, SAFI> to a peer unless the BGP speaker advertises the ADD-PATH Capability to the peer indicating its ability to send multiple paths for the <AFI, SAFI>, and also receives the ADD-PATH Capability from the peer indicating its ability to receive multiple paths for the <AFI, SAFI>..."

Does your implementation follow normal procedures when generating UPDATES if the ADD-PATH capability is not sent and received?

[RFC2119]: MUST

Implementation	Yes/No/Other	Comments
Cumulus	Yes	
Cisco	Yes	
Exa	Yes	

4.4.6. Update Generation with Encoding

Functionality/Description: "...in which case the speaker MUST generate a route update for the <AFI, SAFI> based on the combination of the address prefix and the Path Identifier, and use the extended NLRI encodings specified in this document."

If the ADD-PATH capability has been sent and received, does your implementation generate new UPDATEs using the (prefix, path identifier) pair and the encodings defined in this document?

[RFC2119]: MUST

Implementation	Yes/No/Other	Comments
Cumulus	Other	TX is not supported yet
Cisco	Yes	
Exa	Yes	

4.4.7. Multiple Address Family Support

Functionality/Description: "The peer SHALL act accordingly in processing an UPDATE message related to a particular <AFI, SAFI>."

Does your implementation support the use of the ADD-PATH capability for multiple <AFI, SAFI> pairs?

[RFC2119]: SHALL

Implementation	Yes/No/Other	Comments
Cumulus	Yes	
Cisco	Yes	
Exa	Yes	

4.4.8. Multiple Address Family Support (2)

Functionality/Description: Which <AFI, SAFI> pairs does your implementation support when using the ADD-PATH capability?

[RFC2119]: N/A

Implementation	Comments
Cumulus	IPv4 unicast and IPv6 unicast
Cisco	ipv4 unicast and ipv6 unicast
Exa	1/1 2/1 1/4 2/4

4.4.9. Bestpath

Functionality/Description: "A BGP speaker SHOULD include the bestpath when more than one path are advertised to a neighbor unless the bestpath is a path received from that neighbor."

Does your implementation include the bestpath when multiple paths are announced to a neighbor, as described?

[RFC2119]: SHOULD

Implementation	Yes/No/Other	Comments
Cumulus	Yes	
Cisco	Yes	
Exa	Other	ExaBGP does not have a FIB, this is user controlled.

4.4.10. Path Identifier Persistency

Functionality/Description: "As the Path Identifiers are locally assigned, and may or may not be persistent across a control plane restart of a BGP speaker..."

Are the path identifiers persistent across control plane restarts in your implementation?

[RFC2119]: N/A

Implementation	Yes/No/Other	Comments
Cumulus	No	
Cisco	No	XE-BGP-ADD-Paths need to have HA enhancements
Exa	Other	User controlled

4.4.11. Graceful Restart

Functionality/Description: "...an implementation SHOULD take special care so that the underlying forwarding plane of a "Receiving Speaker" as described in [RFC4724] is not affected during the graceful restart of a BGP session."

Please explain how your implementation addresses Graceful Restart.

[RFC2119]: SHOULD

Implementation	Comments
Cumulus	Quagga has partial GR support (it is GR aware for other restarting nodes) but does not maintain the forwarding plane during a restart.
Cisco	XE-BGP-ADD-Paths need to have HA enhancements
Exa	No FIB, not relevant

4.5. Section 6: Applications

4.5.1. Applications

Functionality/Description: Please list or explain which applications that require the propagation of multiple paths are supported by your implementation.

[RFC2119]: N/A

Implementation	Comments
Cumulus	None yet....RX onlys
Cisco	1. RR client to RR use cases for ipv4 and ipv6. 2. RR to RR clients(could be ASBRs) use cases for ipv4 and ipv6.
Exa	N/A

4.6. Section 7: Deployment Considerations

4.6.1. Deployment Experience

Functionality/Description: Please comment on deployment experience with your implementation.

[RFC2119]: N/A

Implementation	Comments
Cumulus	
Cisco	
Exa	Cisco routers exporting ADD-PATH routes to ExaBGP, routes are then stored in a distributed Database. A complex best path selection (including latency) is performed on the stored routes, and the best routes are then re-injected in the core via ExaBGP.

5. Security Considerations

This document reports the results of an ADD-PATH implementation survey. As such, it does not iintroduce any security risks.

6. IANA Considerations

This document has no IANA actions.

7. Acknowledgements

The editor would like to thank Daniel Walton, Mohammed Mirza and Thomas Mangin.

8. References

8.1. Normative References

[I-D.ietf-idr-add-paths] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", draft-ietf-idr-add-paths-10 (work in progress), October 2014.

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

8.2. Informative References

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

Author's Address

Alvaro Retana
Cisco Systems, Inc.
7025 Kit Creek Rd.
Research Triangle Park, NC 27709
USA

Email: aretana@cisco.com

IDR
Internet-Draft
Intended status: Informational
Expires: April 11, 2015

G. Van de Velde
A. Karch
Cisco Systems
W. Henderickx
Alcatel-Lucent

October 8, 2014

Dissemination of Flow Specification Rules for IPv6 Implementation Report draft-vandevelde-idr-ipv6-flowspec-imp-00

Abstract

This document is an implementation report for the BGP Flow Specification Rules for IPv6 as defined in [I-D.ietf-idr-flow-spec-v6]. The respondents are experts with the implementations they reported on, and their responses are considered authoritative for the implementations for which their responses represent.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 11, 2015.

Table of Contents

1. Introduction

In order to share Flow Specification Rules for IPv6 using the BGP routing protocol a new BGP Network Layer Reachability Information (NLRI) encoding format is required.

This document provides an implementation report for the BGP Dissemination of Flow Specification Rules for IPv6 NLRI Format as defined in [I-D.ietf-idr-flow-spec-v6].

The editors did not verify the accuracy of the information provided by respondents or by any alternative means. The respondents are experts with the implementations they reported on, and their responses are considered authoritative for the implementations for which their responses represent.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC2119] only when they appear in all upper case. They may also appear in lower or mixed case as English words, without any normative meaning.

3. Implementation Forms

Contact and implementation information for person filling out this form:

Cisco
Name: Gunter Van de Velde
Email:gvandeve@cisco.com
Vendor: Cisco Systems, Inc.
Release: IOS-XR
Protocol Role: Sender, Receiver

Alcatel-Lucent
Name: Wim Henderickx
Email: wim.henderickx@alcatel-lucent.com
Vendor: Alcatel-Lucent, Inc.
Release: R12R4
Protocol Role: Sender, Receiver

4. NLRI and Extended Community subtypes

Does the implementation support the Network Layer Reachability (NLRI) subtypes as described in Section 3 and 4 of [I-D.ietf-idr-flow-spec-v6].

- N1: Type 1 - Destination IPv6 Prefix
- N2: Type 2 - Source IPv6 Prefix
- N3: Type 3 - Next Header
- N4: Type 4 - Port
- N5: Type 5 - Destination port
- N6: Type 6 - Source port
- N7: Type 7 - ICMP type
- N8: Type 8 - ICMP code
- N9: Type 9 - TCP flags
- N10: Type 10 - Packet length
- N11: Type 11 - DSCP (Diffserv Code Point)
- N12: Type 12 - Fragment

- N13: Type 13 - Flow Label
- E1: Extended Community - traffic-rate
- E2: Extended Community - traffic-action
- E3: Extended Community - redirect
- E4: Extended Community - traffic-marking

	Cisco	ALU	TBD
Rcv.N1	YES	YES	---
Snd.N1	YES	YES	---
Rcv.N2	YES	YES	---
Snd.N2	YES	YES	---
Rcv.N3	YES	YES	---
Snd.N3	YES	YES	---
Rcv.N4	YES	YES	---
Snd.N4	YES	YES	---
Rcv.N5	YES	YES	---
Snd.N5	YES	YES	---
Rcv.N6	YES	YES	---
Snd.N6	YES	YES	---
Rcv.N7	YES	YES	---
Snd.N7	YES	YES	---
Rcv.N8	YES	YES	---
Snd.N8	YES	YES	---
Rcv.N9	YES	YES	---
Snd.N9	YES	YES	---
Rcv.N10	YES	YES	---
Snd.N10	YES	YES	---
Rcv.N11	YES	YES	---
Snd.N11	YES	YES	---
Rcv.N12	YES	YES	---
Snd.N12	YES	YES	---
Rcv.N13	YES	YES	---
Snd.N13	YES	YES	---
Rcv.E1	YES	YES	---
Snd.E1	YES	YES	---
Rcv.E2	YES	YES	---
Snd.E2	YES	YES	---
Rcv.E3	YES	YES	---
Snd.E3	YES	YES	---
Rcv.E4	YES	YES	---
Snd.E4	YES	YES	---

Yes

- Rcv: BGP speaker can receive the information into the BGP process
- Snd: BGP speaker can relay the information from the BGP process

No

- o Rcv: BGP speaker can not receive the information into the BGP process
- o Snd: BGP speaker can not relay the information from the BGP process

5. Interoperable Implementations

Summary of executed Interop tests between different implementations

5.1. Alcatel-Lucent - Cisco Systems

This Interop test was between a Cisco IOS-XR router and a Alcatel-Lucent Router. Between the two BGP devices an iBGP session is established.

The following IPv6 Flow Specification NLRI is constructed using the Cisco router as IPv6 Flow Specification controller:

```
!
class-map type traffic match-all InteropMatchList
 match destination-address ipv6 2001:2::3/128
 match source-address ipv6 2002:2::3/128
 match destination-port 1-5 7-11 13-18 20-25 27-31
 match source-port 33-37 39-43 45-50 53-58 60-65
 match ipv6 icmp-type 35
 match ipv6 icmp-code 55
 match packet length 120-130 135-140 145-160 165-200 205-225
 match dscp 1-10 11-20 22-30 32-40 52-60
 match tcp-flag 240 any
 match protocol 6-71 73-80 85-90 95-105 110-115
 end-class-map
!
policy-map type pbr InteropCiscoAlu
 class type traffic InteropMatchList
  police rate 200 bps
  !
  redirect nexthop 2001::1
  set dscp 45
 !
 class type traffic class-default
 !
 end-policy-map
```

This results with the following Flow Specification Extended communities and IPv6 Flow Specification NLRI:

```
            AFI: IPv6
             NLRI (Hex dump) :
            0x018000200100020000000000000000000000030280002002000200000000000
            000000000000303030645470349455003554 55a035f4569036ec5730503014505
            0307450b030d45120314451903 1bc51f06032145250327452b032d45320335453
            a033cc5410781230881370980f00a037845820387458c039145a003a545c803cd
            c5e10b0301450a030b45140316451e032045280334c53c
             Actions       :Traffic-rate: 200 bps DSCP: 45
               Nexthop: 2001::1  (policy.1.test1)
```

The above IPv6 Flow Specification rule is correctly received by the Alcatel-Lucent BGP speaker and is reflected as follows on the device:

```
*A:PE26>config>service>vprn>sub-if>grp-if>sap>static-host# show router 117 bgp
routes flow-ipv6
===============================================================================
 BGP Router ID:195.207.5.200    AS:65117       Local AS:65117
===============================================================================
 Legend -
 Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
 Origin codes  : i - IGP, e - EGP, ? - incomplete, > - best, b - backup

===============================================================================
BGP FLOW IPV6 Routes
===============================================================================
Flag  Network             Nexthop                 LocalPref       MED
      As-Path
-------------------------------------------------------------------------------
u*>i  --                  2001::1                 100             None
      No As-Path

      Community Action:    ext:800:0
      Community Action:    rate-limit: 65117:110*
      Community Action:    mark-dscp: 45
      NLRI Subcomponents:
      Dest Pref : 2001:2::3/128 offset 0
      Src Pref  : 2002:2::3/128 offset 0
      Ip Proto  : [ >= 6 ] and [ <= 71 ] or [ >= 73 ] and [ <= 80 ] or [ >=
      Dest Port : [ >= 1 ] and [ <= 5 ] or [ >= 7 ] and [ <= 11 ] or [ >= 13
      Src Port  : [ >= 33 ] and [ <= 37 ] or [ >= 39 ] and [ <= 43 ] or [ >=
      ICMP Type : [ == 35 ]
      ICMP Code : [ == 55 ]
      TCP Flags : [ 240 ]
      TCP Flags : [ 240 ]
      DSCP      : [ >= 1 ] and [ <= 10 ] or [ >= 11 ] and [ <= 20 ] or [ >=
-------------------------------------------------------------------------------
Routes : 1
```

6. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: The IANA has requested that this section remain in the document upon publication as an RFC. This note to the RFC Editor, however, may be removed.

7. Security Considerations

No new security issues are introduced to the BGP defined in Dissemination of Flow Specification Rules for IPv6 [I-D.ietf-idr-flow-spec-v6].

8. Privacy Considerations

No new privacy issues are introduced to the BGP defined in Dissemination of Flow Specification Rules for IPv6 [I-D.ietf-idr-flow-spec-v6].

9. Acknowledgements

The authors would like to thank Nicolas Fevrier, Hyojeong Kim, Bertrand Duvivier and Adam Simpson.

10. Change Log

Initial Version: 8 October 2014

11. References

11.1. Normative References

[I-D.ietf-idr-flow-spec-v6] Raszuk, R., Pithawala, B., McPherson, D., and A. Andy, "Dissemination of Flow Specification Rules for IPv6", draft-ietf-idr-flow-spec-v6-05 (work in progress), March 2014.

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

11.2. Informative References

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

Authors' Addresses

Gunter Van de Velde
Cisco Systems
De Kleetlaan 6a
Diegem 1831
Belgium

Phone: +32 2704 5473
Email: gvandeve@cisco.com

Andy Karch
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95124 95134
USA

Email: akarch@cisco.com

Wim Henderickx
Alcatel-Lucent

Email: wim.henderickx@alcatel-lucent.be

IDR Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 24, 2015

Z. Wang
Q. Wu
Huawei

September 20, 2014

Distribution of MPLS-TE Extended admin Group Using BGP
draft-wang-idr-eag-distribution-00

Abstract

As MPLS-TE network grows, administrative Groups advertised as a fixed-length 32-bit Bitmask is quite constraining. "Extended Administrative Group" IGP TE extensions sub-TLV defined in [I-D.ietf-mpls-extended-admin-group] is introduced to provide for additional administrative groups (link colors) beyond the current limit of 32. This document describes extensions to BGP protocol, that can be used to distribute extended administrative groups in MPLS-TE.

Status of This Memo

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

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

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

This Internet-Draft will expire on March 24, 2015.

Copyright Notice

Table of Contents

1. Introduction

MPLS-TE advertises 32 administrative groups (commonly referred to as "colors" or "link colors") using the Administrative Group sub-TLV of the Link TLV defined in OSPFv2 (RFC3630), OSPFv3 (RFC5329) and ISIS (RFC5305).

As MPLS-TE network grows, administrative Groups advertised as a fixed-length 32-bit Bitmask is quite constraining. "Extended Administrative Group" IGP TE extensions sub-TLV defined in [I-D.ietf-mpls-extended-admin-group] is introduced to provide for additional administrative groups (link colors) beyond the current limit of 32.

This document proposes new BGP Link attribute TLVs that can be announced as attribute in the BGP-LS attribute (defined in [I.D-ietf-idr-ls-distribution]) to distribute extended administrative groups in MPLS-TE.

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 [RFC2119].

3. Carrying Extended Administrative Groups in BGP

This document proposes one new BGP link attribute TLVs that can be announced as attribute in the BGP-LS attribute (defined in [I.D-ietf-idr-ls-distribution]) to distribute extended administrative groups. The extensions in this document build on the ones provided in BGP-LS [I.D-ietf-idr-ls-distribution] and BGP-4 [RFC4271].

BGP-LS attribute defined in [I.D-ietf-idr-ls-distribution] has nested TLVs which allow the BGP-LS attribute to be readily extended. Link attribute TLVs defined in section 3.2.2 of [I-D.ietf-idr-ls-distribution]are TLVs that may be encoded in the BGP-LS attribute with a link NLRI. Each 'Link Attribute' is a Type/Length/ Value (TLV) triplet formatted as defined in Section 3.1 of [I-D.ietf-idr-ls-distribution].

This document proposes one new TLV as a link attribute:

Type	Value
TBD1	Extended Admin Group (EAG)

The EAG TLV is used in addition to the Administrative Groups when a node wants to advertise more than 32 colors for a link. The EAG TLV is optional. The format and semantics of the 'value' fields in EAG TLVs correspond to the format and semantics of value fields in IGP extension sub-TLVs, defined in [I-D.ietf-mpls-extended- admin-group].

TLV Code Point	Description	IS-IS TLV/Sub-TLV	Defined in:
xxxx	Extended Admin Group	22/xx	[I-D.ietf-mpls-extended-admin-group

Table 1: 'EAG' Link Attribute TLV

3.1. AG and EAG coexistence

Similar to section 2.3.1 of [I-D.ietf-mpls-extended- admin-group],if a BGP speaker advertises both AG and EAG then AG and EAG should be dealt with in the same way as AG and EAG carried in the Extended Administrative Group (EAG) sub-TLV [I-D.ietf-mpls-extended- admin-group] for both OSPF [RFC3630] and ISIS [RFC5305].

3.2. Desire for unadvertised EAG bits

Unlike AGs, EAGs are advertised as any non-zero-length-bit Bitmask. the EAG length may be longer for some links than for others. Similar to section 2.3.2 of [I-D.ietf-mpls-extended- admin-group], if a BGP peer wants to only use links where the specific bits of an EAG is set to 1 but the specific bits of this EAG is not advertised, then the implementation SHOULD process these desire and unadvertised EAG bits

in accordance with rule defined in section 2.3.2 of [I.D-ietf-mpls-extended-admin- group].

4. Security Considerations

This document does not introduce security issues beyond those discussed in [I.D-ietf-idr-ls-distribution] and [RFC4271].

5. IANA Considerations

IANA maintains the registry for the TLVs. BGP Extended Admin Group link attribute TLV will require one new type code defined in this document.

6. Acknowledgments

The authors gratefully acknowledge the review made by Eric Osborne.

7. Normative References

[I-D.ietf-idr-ls-distribution]
Gredler, H., "North-Bound Distribution of Link-State and TE Information using BGP", ID draft-ietf-idr-ls-distribution-03, May 2013.

[I-D.ietf-mpls-extended-admin-group]
Osborne, E., "Extended Administrative Groups in MPLS-TE", ID draft-ietf-mpls-extended-admin-group-07, May 2014.

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

[RFC3630] Katz, D., Yeung, D., and K. Kompella, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.

[RFC4271] Rekhter, Y., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.

Authors' Addresses

Zitao Wang
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: wangzitao@huawei.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 4, 2015

A. Zhdankin
K. Patel
A. Clemm
Cisco

October 1, 2014

Yang Data Model for BGP Protocol
draft-zhdankin-netmod-bgp-cfg-01.txt

Abstract

This document defines a YANG data model that can be used to configure and manage BGP.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 4, 2015.

Copyright Notice

Table of Contents

1. Introduction

YANG [RFC6020] is a data definition language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces (e.g. ReST) and encodings other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of

implementation for other interfaces, such as CLI and programmatic APIs.

This document defines a YANG data model that can be used to configure and manage BGP. The data model is very comprehensive in scope, resulting in a very large module being defined. When contemplating whether it would be appropriate to introduce a data model of such a large scope, we decided that there would be value in particular because BGP defines such a rich set of features, which makes the problem arising from heterogeneity involved when managing these features quite pronounced. Also, there is very little information that is designated as "mandatory", leaving the decision which capabilities to actually support to product implementations.

There are several distinct parts of the data model. The first part, by far the largest, serves to configure and manage BGP itself. It defines a large set of control knobs for that purpose, as well as a few data nodes that can be used to monitor health and gather statistics. The second part, much smaller than the first, defines a data model for the configuration of AS-Path and prefix-based filter lists, in essence policies that define the exchange of BGP messages between BGP peers. Together they form a complete data model that serves as a framework for configuration and management of BGP protocol and its policies.

The YANG module defined in this document has all the common building blocks for BGP protocol namely: Neighbor List, Address Family specific Parameters, Protocol Bestpath specific Parameters, Prefix based Filter Lists, and AS-PATH based Filter Lists.

1.1. Requirements Language

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

2. Definitions and Acronyms

AF: Address Family

AS: Autonomous System

BGP: Border Gateway Protocol

HTTP: Hyper-Text Transfer Protocol

JSON: JavaScript Object Notation

L2VPN: Layer 2 VPN

NETCONF: Network Configuration Protocol

NSAP: Network Service Access Point

ReST: Representational State Transfer, a style of stateless interface and protocol that is generally carried over HTTP

RPKI: Resource Public Key Infrastructure

RTFilter: Route Filter

VPN: Virtual Private Network

YANG: A data definition language for NETCONF

3. The Design of the Core Routing Data Model

3.1. Overview

The overall data model consists of two main components, each contained in its own separate container. Container "bgp-router" is used to configure and manage BGP itself. It is by far the largest part of the model. Container "prefix-lists" is used to configure BGP prefix lists, defining the rules and policies as which BGP information to share with which other nodes.

3.2. BGP Router Configuration

The overall structure of the "bgp-router" part of the model is depicted in the following diagram. Brackets enclose list keys, "rw" means configuration data, "?" designates optional nodes. The figure does not depict all definitions; it is intended to illustrate the overall structure.

```
module: bgp
   +--rw bgp-router
   |  +--rw local-as-number?       uint32
   |  +--rw local-as-identifier?   inet:ip-address
   |  +--rw rpki-config
   |  |  .....
   |  +--rw af-configuration
   |     .....
   +--rw bgp-neighbors
      .....
```

The key components of the "bgp-router" model concern the configuration of the BGP neighbors, of the Resource Public Key Infrastructure (RPKI), and of address families (AF). Each is defined in the following subsections.

3.2.1. AF Configuration

AF-configuration is used to configure and manage BGP configuration on an address family basis. BGP is designed to carry routing information for multiple different address families as specified in [RFC4760]. AF-Configuration is indexed by (router-AS, AFI, SAFI, VRFID) [RFC4760] and [RFC4364]. It contains any AF specific protocol configuration, BGP Bestpath configuration parameters,BGP neighbor configuration parameters, BGP dampening parameters, BGP route aggregation parameters, and any BGP policy configuration like redistribution.

The overall structure of the AF Configuration data model is depicted in the following diagram. As before, brackets enclose list keys, "rw" means configuration data, "?" designates optional nodes, parantheses indicate choices. The figure does not depict all definitions; it is intended to illustrate the overall model structure. Roughly speaking, address family configuration allows for separate configuration of IPv4, IPv6, L2VPN, NSAP, VPNv4 and VPNv6 address families, as well as route filters. Within each address family, you have additional substructure, for example, to distinguish between configuration of unicast and multicast.

```
module: bgp
   +--rw bgp-router
   |  .....
   |  +--rw af-configuration
   |     +--rw ipv4
   |     |  +--rw mdt
   |     |  | .....
   |     |  +--rw multicast
   |     |  |  +--rw bgp
   |     |  |  |  .....
   |     |  |  +--rw auto-summary?        boolean
   |     |  |  +--rw aggregate-address?   inet:ip-address
   |     |  |  +--rw distance?            uint8
   |     |  |  +--rw network?             inet:ip-address
   |     |  |  +--rw (protocol)?
   |     |  |  |  .....
   |     |  |  +--rw default-metric?      uint32
   |     |  +--rw unicast
   |     |  |  +--rw bgp
   |     |  |  |  .....
```

```
|     |  |  +--rw auto-summary?          boolean
|     |  |  +--rw aggregate-address?     inet:ip-address
|     |  |  +--rw distance?              uint8
|     |  |  +--rw network?               inet:ip-address
|     |  |  +--rw (protocol)?
|     |  |  |  .....
|     |  |  +--rw number-of-path?        uint8
|     |  |  +--rw ibgp-number-of-path?   uint8
|     |  |  +--rw synchronization?       boolean
|     |  +--rw mvpn
|     |     +--rw bgp
|     |     |  .....
|     |     +--rw auto-summary?   boolean
|     +--rw ipv6
|     |  +--rw multicast
|     |  |  +--rw bgp
|     |  |  |  .....
|     |  |  +--rw aggregate-address?   inet:ip-address
|     |  |  +--rw distance?            uint8
|     |  |  +--rw network?             inet:ip-address
|     |  |  +--rw (protocol)?
|     |  |     .....
|     |  +--rw unicast
|     |  |  +--rw bgp
|     |  |  |  .....
|     |  |  +--rw aggregate-address?     inet:ip-address
|     |  |  +--rw distance?              uint8
|     |  |  +--rw network?               inet:ip-address
|     |  |  +--rw (protocol)?
|     |  |  |  .....
|     |  |  +--rw default-metric?        uint32
|     |  |  +--rw number-of-path?        uint8
|     |  |  +--rw ibgp-number-of-path?   uint8
|     |  |  +--rw synchronization?       boolean
|     |  +--rw mvpn
|     |     .....
|     +--rw l2vpn
|     |  +--rw vpls
|     |     .....
|     +--rw nsap
|     |  +--rw unicast
|     |     +--rw bgp
|     |     |  .....
|     |     +--rw default-metric?        uint32
|     |     +--rw number-of-path?        uint8
|     |     +--rw ibgp-number-of-path?   uint8
|     |     +--rw network?               inet:ip-address
|     |     +--rw (protocol)?
```

```
|     |     |  .....
|     |     +--rw synchronization?       boolean
|     +--rw rtfilter
|     |  +--rw unicast
|     |     .....
|     +--rw vpnv4
|     |  +--rw unicast
|     |  |  +--rw bgp
|     |  |  |  .....
|     |  |  +--rw number-of-path?        uint8
|     |  |  +--rw ibgp-number-of-path?   uint8
|     |  +--rw multicast
|     |     +--rw bgp
|     |     |  .....
|     |     +--rw number-of-path?        uint8
|     |     +--rw ibgp-number-of-path?   uint8
|     +--rw vpnv6
|        +--rw unicast
|           +--rw bgp
|              .....
```

The key AF configuration components are described in the following subsections.

3.2.1.1. AF Specific Protocol Configuration

AF specific protocol configuration involves configuration of the parameters that are specific to a given AF. For instance, configuration parameters specific to the consistency checking between prefixes and labels are specific to address families that are enabled with Labels. Similarly redistribution of routes from other protocols is specific to Address Families that are supported in other protocols.

3.2.1.2. BGP Bestpath Configuration

BGP BestPath Configuration Parameters involves configuration of the parameters that influence the BGP Bestpath decision. For instance, the ignore-as-path command allows BGP process to ignore as-path length check. The ignore-routerid command allows BGP process to ignore routerid check. The ignore-igp-metric command allows BGP process to ignore igp metric check. The ignore-cost-community command allows BGP process to ignore cost communities. The MED related commands influence MED comparision in the BGP Bestpath decision.

3.2.1.3. BGP Neighbor Configuration

BGP Neighbor Configuration Parameters involves configuration of the parameters that are neighbor address family specific. These commands include neighbor capabilities, neighbor policies and any protocol related parameters that are specific to BGP neighbor.

3.2.1.4. BGP Dampening

BGP Dampening Parameters involves configuration of the parameters that influence BGP Route Dampening. These parameters allow enabling of Route Dampening on an address family level. The Dampening configuration also allows configuration of Dampening specific parameters like max suppress time, resuse threshold, half life, and the suppress threshold.

3.2.1.5. BGP Route Aggregation

BGP Route Aggregation Parameters involves configuration of the parameters that enables BGP Route Aggregation.

3.2.1.6. BGP Redistribution

BGP Route Redistribution Parameters involves configuration of the parameters that enables BGP Route Redistribution from and to the BGP protocol.

3.2.2. BGP Neighbor Configuration

Bgp-neighbor is used to configure and manage BGP neighbors. BGP neighbor configuration is indexed by af-configuration, neighbor address and neighbor-AS. It contains configuration for any policies that are configured for a neighbor on an inbound or an outbound, any transport related configuration parameters, any protocol related configuration parameters, and any protocol capabilities related configuration parameters.

The following diagram depicts the overall structure of the BGP Neighbors subtree. Brackets enclose list keys, "rw" means configuration, "ro" operational state data, and "?" designates optional nodes. Parantheses enclose choice and case nodes. The figure does not depict all definitions; it is intended to illustrate the overall structure.

```
module: bgp
   + ....
   +--rw bgp-neighbors
   |  +--rw bgp-neighbor [as-number]
```

```
|     +--rw as-number                   uint32
|     +--rw (peer-address-type)?
|     |  .....
|     +--rw prefix-list?                prefix-list-ref
|     +--rw default-action?             actions-enum
|     +--rw af-specific-config
|     |  +--rw ipv4
|     |  |  +--rw mdt
|     |  |  |  .....
|     |  |  +--rw unicast
|     |  |  |  .....
|     |  |  +--rw multicast
|     |  |  |  .....
|     |  |  +--rw mvpn
|     |  |     .....
|     |  +--rw ipv6
|     |  |  +--rw unicast
|     |  |  |  .....
|     |  |  +--rw multicast
|     |  |  |  .....
|     |  |  +--rw mvpn
|     |  |     .....
|     |  +--rw l2vpn
|     |  |  +--rw evpn
|     |  |  |  .....
|     |  |  +--rw vpls
|     |  |     .....
|     |  +--rw nsap
|     |  |  +--rw unicast
|     |  |     .....
|     |  +--rw rtfilter
|     |  |  +--rw unicast
|     |  |     .....
|     |  +--rw vpnv4
|     |  |  +--rw unicast
|     |  |  |  .....
|     |  |  +--rw multicast
|     |  |     .....
|     |  +--rw vpnv6
|     |     +--rw unicast
|     |     |  .....
|     |     +--rw multicast
|     |        .....
|     +--rw bgp-neighbor-state
|     |  .....
|     +--rw bgp-neighbor-statistics
|        .....
```

3.2.3. BGP RPKI

rpki-config is used to configure and manage BGP Origin Validation. This feature is specific to IPv4 and IPv6 Address Families. It is indexed by af-configuration. It contains the configuration commands for the BGP RPKI Server, RPKI RTR Protocol and the BGP protocol. This includes configuration for the Server address, Server preference, RPKI RTR protocol specific parameters, choice of a transport for RPKI RTR Protocol, and BGP specific parameters including enabling and disabling of this feature for IBGP and EBGP routes.

The structure of the RPKI configuration data model is depicted below, per the same conventions used in the earlier diagrams.

```
module: bgp
   +--rw bgp-router
   |  .....
   |  +--rw rpki-config
   |  |  +--rw cache-server-config
   |  |  |  .....
   |  |  +--rw validation-config
   |  |  |  .....
   |  |  +--rw bestpath-computation
   |  |     .....
```

3.3. Prefix Lists

BGP Prefix Lists are used to manipulate Prefix information carried within a BGP. The prefix information carried within BGP is filtered or allowed using BGP Prefix Lists. BGP Prefix Lists consists of an ordered set of one or more rules that describe IPv4 or IPv6 prefixes range and an associated action rule that describes whether the matching prefixes should be dropped or permitted. The Prefix Lists are usually applied to a BGP neighbor as part of an inbound policy (applied to prefixes received by a neighbor) or an outbound policy (applied to prefixes sent by a neighbor).

The structure of the prefix list configuration data model is depicted below, per the same conventions used in the earlier diagrams.

```
   module: bgp
      .....
      +--rw prefix-lists
         +--rw prefix-list [prefix-list-name]
            +--rw prefix-list-name    string
            +--rw prefixes
               +--rw prefix [seq-nr]
                  +--rw seq-nr           uint16
                  +--rw prefix-filter
                     +--rw (ip-address-group)?
                     |  .....
                     +--rw action             actions-enum
                     +--rw statistics
                        .....
```

Prefix lists are defined in a list in a designated container. Each prefix list in turn contains a list of prefixes, indexed by a sequency number. Each prefix is comprised of a prefix filter, used to match BGP packets, an action that is applied when a filter matches, and a set of statistics that indicate how often individual prefixes are applied.

4. BGP Yang Module

<CODE BEGINS> file "bgp@2013-07-15.yang"

```
module bgp {
  namespace "urn:cisco:params:xml:ns:yang:bgp";
  // replace with IANA namespace when assigned
  prefix bgp;

  import ietf-inet-types {
    prefix inet;
  }
  import ietf-yang-types {
    prefix yang;
  }

  organization
    "Cisco Systems
     170 West Tasman Drive
     San Jose, CA 95134-1706
     USA";
  contact
    "Aleksandr Zhdankin azhdanki@cisco.com
     Keyur Patel keyupate@cisco.com
     Alexander Clemm alex@cisco.com";
```

```
  description
    "This YANG module defines the generic configuration
     data for BGP, which is common across all of the vendor
     implementations of the protocol. It is intended that the module
     will be extended by vendors to define vendor-specific
     BGP configuration parameters and policies,
     for example route maps or route policies.


     Terms and Acronyms

     BGP (bgp): Border Gateway Protocol

     IP (ip): Internet Protocol

     IPv4 (ipv4):Internet Protocol Version 4

     IPv6 (ipv6): Internet Protocol Version 6

     MED(med): Multi Exit Discriminator

     IGP (igp): Interior Gateway Protocol

     MTU (mtu) Maximum Transmission Unit
    ";

  revision 2013-07-15 {
    description
      "Initial revision.";
  }

  typedef prefix-list-ref {
    description
      "A reference to the prefix list which a bgp-neighbor can use.";
    type leafref {
      path "/prefix-lists/prefix-list/prefix-list-name";
    }
  }

  typedef neigbour-ref {
    description
      "A reference to the bgp-neighbor.";
    type leafref {
      path "/bgp-neighbors/bgp-neighbor/as-number";
    }
  }

  typedef bgp-peer-admin-status {
```

```
    description
      "Administartive status of a BGP peer.";
    type enumeration {
      enum "unknown";
      enum "up";
      enum "down";
    }
  }

  typedef actions-enum {
    description
      "Permit/deny action.";
    type enumeration {
      enum "permit";
      enum "deny";
    }
  }

  grouping ACTIONS {
    description
      "Permit/deny action.";
    leaf action {
      type actions-enum;
      mandatory true;
    }
  }

  grouping slow-peer-config {
    description
      "Configure a slow-peer.";
    container detection {
      leaf enable {
        type boolean;
        default "true";
      }
      leaf threshold {
        type uint16 {
          range "120..3600";
        }
      }
    }
    leaf split-update-group {
      type enumeration {
        enum "dynamic";
        enum "static";
      }
    }
  }
```

```
  grouping update-group-management {
    description
      "Manage peers in BGP update group.";
    leaf split-as-override {
      description
        "Keeps peers with as-override in different update groups.";
      type boolean;
    }
  }

  grouping neighbour-base-af-config {
    description
      "A set of configuration parameters that is applicable to all neighbour add
ress families.";
    leaf active {
      description
        "Enable the address family for this neighbor.";
      type boolean;
      default "false";
    }
    leaf advertisement-interval {
      description
        "Minimum interval between sending BGP routing updates.";
      type uint32;
    }
    leaf allowas-in {
      description
        "Accept as-path with my AS present in it.";
      type boolean;
      default "false";
    }
    leaf maximum-prefix {
      description
        "Maximum number of prefixes accepted from this peer.";
      type uint32;
    }
    leaf next-hop-self {
      description
        "Enable the next hop calculation for this neighbor.";
      type boolean;
      default "true";
    }
    leaf next-hop-unchanged {
      description
        "Propagate next hop unchanged for iBGP paths to this neighbour.";
      type boolean;
      default "true";
    }
    container remove-private-as {
```

```
      leaf remove-private-as-number {
        description
          "Remove private AS number from outbound updates.";
        type boolean;
      }
      leaf replace-with-local-as {
        description
          "Replace private AS number with local AS.";
        type boolean;
      }
    }
    leaf route-reflector-client {
      description
        "Configure a neighbor as Route Reflector client.";
      type boolean;
      default "false";
    }
    leaf send-community {
      description
        "Send Community attribute to this neighbor.";
      type enumeration {
        enum "both";
        enum "extended";
        enum "standard";
      }
      default "standard";
    }
    uses slow-peer-config;
    leaf soo {
      description
        "Site-of-Origin extended community. Format is ASN:nn or IP-address:nn";
      type string;
    }
    leaf weight {
      description
        "Set default weight for routes from this neighbor.";
      type uint16;
    }
  }

  grouping neighbour-common-af-config {
    description
      "A set of configuration parameters that is applicable to all neighbour add
ress families,
       except of nsap and rtfilter.";
    uses neighbour-base-af-config;
    leaf prefix-list {
      description
        "Reference to the prefix list of this neighbour.";
```

```
      type prefix-list-ref;
    }
    leaf soft-reconfiguration {
      description
        "Allow inbound soft reconfiguration.";
      type boolean;
    }
  }

  grouping neighbour-cast-af-config {
    description
      "A set of configuration parameters that is applicable to both unicast and
multicast sub-address families.";
    uses neighbour-common-af-config;
    leaf propagate-dmzlink-bw {
      description
        "Propagate the DMZ link bandwidth.";
      type boolean;
    }
    container default-originate {
      description
        "Originate default route to this neighbor.";
      leaf enable {
        type boolean;
        default "false";
      }
    }
  }

  grouping neighbour-ip-multicast-af-config {
    description
      "A set of configuration parameters that is applicable to ip multicast.";
    uses neighbour-cast-af-config;
    leaf route-server-client-context {
      description
        "Specifies Route Server client context name.";
      type string;
    }
  }

  grouping neighbour-ip-unicast-af-config {
    description
      "A set of configuration parameters that is applicable to ip unicast.
       This grouping is intended to be extended by vendors as necessary to descr
ibe the vendor-specific configuration parameters.";
    uses neighbour-ip-multicast-af-config;
  }

  grouping bgp-af-config {
    description
```

```
      "A set of configuration parameters that is applicable to all address famil
ies of the BFP router.";
    leaf additional-paths {
      description
        "Additional paths in the BGP table.";
      type enumeration {
        enum "all";
        enum "best-n";
        enum "group-best";
      }
    }
    leaf advertise-best-external {
      description
        "Advertise best external path to internal peers.";
      type boolean;
    }
    container aggregate-timer {
      description
        "Configure aggregation timer.";
      leaf enable {
        type boolean;
        default "true";
      }
      leaf threshold {
        type uint16 {
          range "6..60";
        }
      }
    }
    container bestpath {
      description
        "Change the default bestpath selection.";
      choice bestpath-selection {
        case as-path {
          description
            "Configures a BGP routrer to not consider the autonomous system (AS)
 path during best path route selection.";
          leaf ignore-as-path {
            type boolean;
            default "false";
          }
        }
        case compare-routerid {
          description
            "Configures a BGP routrer to compare identical routes received from
different external peers
             during the best path selection process and to select the route with
 the lowest router ID as the best path.";
          leaf ignore-routerid {
            type boolean;
            default "false";
          }
```

```
        }
        case cost-community {
          description
            "Configures a BGP router to not evaluate the cost community attribut
e
             during the best path selection process.";
          leaf ignore-cost-community {
            type boolean;
            default "false";
          }
        }
        case igp-metric {
          description
            "Configures the system to ignore the IGP metric during BGP best path
 selection.";
          leaf ignore-igp-metric {
            type boolean;
            default "false";
          }
        }
        case mad-confed {
          description
            "Configure a BGP routing process to compare the Multi Exit Discrimin
ator (MED)
             between paths learned from confederation peers.";
          leaf enable {
            type boolean;
            default "false";
          }
          leaf missing-as-worst {
            description
              "Assigns a value of infinity to routes that are missing
               the Multi Exit Discriminator (MED) attribute,
               making the path without a MED value the least desirable path";
            type boolean;
            default "false";
          }
        }
      }
    }
    leaf dampening {
      description
        "Enable route-flap dampening.";
      type boolean;
      default "false";
    }
    leaf propagate-dmzlink-bw {
      description
        "Use DMZ Link Bandwidth as weight for BGP multipaths.";
      type boolean;
    }
```

```
    leaf redistribute-internal {
      description
        "Allow redistribution of iBGP into IGPs (dangerous)";
      type boolean;
    }
    leaf scan-time {
      description
        "Configure background scanner interval in seconds.";
      type uint8 {
        range "5..60";
      }
    }
    uses slow-peer-config;
    leaf soft-reconfig-backup {
      description
        "Use soft-reconfiguration inbound only when route-refresh is not negotia
ted.";
      type boolean;
    }
  }

  grouping bgp-af-vpn-config {
    description
      "A set of configuration parameters that is applicable to vpn sub-address f
amily on the BGP router.";
    uses bgp-af-config;
    uses update-group-management;
  }

  grouping bgp-af-mvpn-config {
    description
      "A set of configuration parameters that is applicable to mvpn sub-address
family on the BGP router.";
    leaf scan-time {
      description
        "Configure background scanner interval in seconds.";
      type uint8 {
        range "5..60";
      }
    }
    uses slow-peer-config;
    leaf soft-reconfig-backup {
      description
        "Use soft-reconfiguration inbound only when route-refresh is not negotia
ted.";
      type boolean;
    }
    leaf propagate-dmzlink-bw {
      description
        "Use DMZ Link Bandwidth as weight for BGP multipaths.";
      type boolean;
    }
```

```
    leaf rr-group {
      description
        "Extended community list name.";
      type string;
    }
    uses update-group-management;
  }

  grouping redistribute {
    description
      "Redistribute information from another routing protocol.
       This grouping is intended to be augmented by vendors to implement vendor-
specific protocol redistribution configuration options.";
    choice protocol {
      case bgp {
        leaf enable-bgp {
          type boolean;
        }
      }
      case ospf {
        leaf enable-ospf {
          type boolean;
        }
      }
      case isis {
        leaf enable-isis {
          type boolean;
        }
      }
      case connected {
        leaf enable-connected {
          type boolean;
        }
      }
      case eigrp {
        leaf enable-eigrp {
          type boolean;
        }
      }
      case mobile {
        leaf enable-mobile {
          type boolean;
        }
      }
      case static {
        leaf enable-static {
          type boolean;
        }
      }
```

```
      case rip {
        leaf enable-rip {
          type boolean;
        }
      }
    }
  }

  grouping router-af-config {
    description
      "A set of configuration parameters that is applicable to all address famil
ies on the BGP router.";
    leaf aggregate-address {
      description
        "Configure BGP aggregate address.";
      type inet:ip-address;
    }
    leaf distance {
      description
        "Define an administrative distance.";
      type uint8 {
        range "1..255";
      }
    }
    leaf network {
      description
        "Specify a network to announce via BGP.";
      type inet:ip-address;
    }
    uses redistribute;
  }

  grouping maximum-paths {
    description
      "Configures packet forwarding over multiple paths.";
    leaf number-of-path {
      type uint8 {
        range "1..32";
      }
    }
    leaf ibgp-number-of-path {
      type uint8 {
        range "1..32";
      }
    }
  }

  container bgp-router {
    description
```

```
      "This is a top-level container for the BGP router.";
    leaf local-as-number {
      type uint32;
    }
    leaf local-as-identifier {
      type inet:ip-address;
    }
    container rpki-config {
      description
        "RPKI configuration parameters.";
      container cache-server-config {
        description
          "Configure the RPKI cache-server parameters in rpki-server configurati
on mode.";
        choice server {
          case ip-address {
            leaf ip-address {
              type inet:ip-address;
              mandatory true;
            }
          }
          case host-name {
            leaf ip-host-address {
              type inet:host;
              mandatory true;
            }
          }
        }
        choice transport {
          description
            "Specifies a transport method for the RPKI cache.";
          case tcp {
            leaf tcp-port {
              type uint32;
            }
          }
          case ssh {
            leaf ssh-port {
              type uint32;
            }
          }
        }
        leaf user-name {
          type string;
        }
        leaf password {
          type string;
        }
        leaf preference-value {
```

```
          description
            "Specifies a preference value for the RPKI cache.
             Setting a lower preference value is better.";
          type uint8 {
            range "1..10";
          }
        }
        leaf purge-time {
          description
            "Configures the time BGP waits to keep routes from a cache after the
 cache session drops. Set purge time in seconds.";
          type uint16 {
            range "30..360";
          }
        }
        choice refresh-time {
          description
            "Configures the time BGP waits in between sending periodic serial qu
eries to the cache. Set refresh-time in seconds.";
          case disable {
            leaf refresh-time-disable {
              type boolean;
            }
          }
          case set-time {
            leaf refresh-interval {
              type uint16 {
                range "15..3600";
              }
            }
          }
        }
        choice responce-time {
          description
            "Configures the time BGP waits for a response after sending a serial
 or reset query. Set response-time in seconds.";
          case disable {
            leaf responce-time-disable {
              type boolean;
            }
          }
          case set-time {
            leaf responce-interval {
              type uint16 {
                range "15..3600";
              }
            }
          }
        }
      }
      container validation-config {
```

```
        description
          "Controls the behavior of RPKI prefix validation processing.";
        leaf enable {
          description
            "Enables RPKI origin-AS validation.";
          type boolean;
          default "true";
        }
        leaf enable-ibgp {
          description
            "Enables the iBGP signaling of validity state through an extended-co
mmunity.";
          type boolean;
        }
        choice validation-time {
          description
            "Sets prefix validation time (in seconds) or to set off the automati
c prefix validation after an RPKI update.";
          case validation-off {
            leaf disable {
              type boolean;
            }
          }
          case set-time {
            leaf prefix-validation-time {
              description
                "Range in seconds.";
              type uint16 {
                range "5..60";
              }
            }
          }
        }
      }
      container bestpath-computation {
        description
          "Configures RPKI bestpath computation options.";
        leaf enable {
          description
            "Enables the validity states of BGP paths to affect the path's prefe
rence in the BGP bestpath process.";
          type boolean;
        }
        leaf allow-invalid {
          description
            "Allows all 'invalid' paths to be considered for BGP bestpath comput
ation.";
          type boolean;
        }
      }
    }
    container af-configuration {
```

```
      description
        "Top level container for address families specific configuration of the
BGP router.";
      container ipv4 {
        container mdt {
          container bgp {
            description
              "BGP specific commands for ipv4-mdt address family/sub-addess fami
ly combination.";
            leaf dampening {
              description
                "Enable route-flap dampening.";
              type boolean;
              default "false";
            }
            leaf scan-time {
              description
                "Configure background scanner interval in seconds.";
              type uint8 {
                range "5..60";
              }
            }
            uses slow-peer-config;
            leaf soft-reconfig-backup {
              description
                "Use soft-reconfiguration inbound only when route-refresh is not
 negotiated.";
              type boolean;
            }
            leaf propagate-dmzlink-bw {
              description
                "Use DMZ Link Bandwidth as weight for BGP multipaths.";
              type boolean;
            }
          }
        }
        container multicast {
          container bgp {
            description
              "BGP specific commands for ipv4-multicast address family/sub-addes
s family combination.";
            uses bgp-af-config;
          }
          leaf auto-summary {
            description
              "Enable automatic network number summarization";
            type boolean;
          }
          uses router-af-config;
          leaf default-metric {
            description
              "Set metric of redistributed routes.";
```

```
            type uint32;
          }
        }
        container unicast {
          container bgp {
            description
              "BGP specific commands for ipv4-unicast address family/sub-addess
family combination.";
            uses bgp-af-config;
            leaf always-compare-med {
              description
                "Allow comparing MED from different neighbors.";
              type boolean;
              default "false";
            }
            leaf enforce-first-as {
              description
                "Enforce the first AS for EBGP routes(default).";
              type boolean;
              default "true";
            }
            leaf fast-external-fallover {
              description
                "Immediately reset session if a link to a directly connected ext
ernal peer goes down.";
              type boolean;
              default "true";
            }
            leaf suppress-inactive {
              description
                "Suppress routes that are not in the routing table.";
              type boolean;
            }
            leaf asnotation {
              description
                "Sets the default asplain notation.";
              type enumeration {
                enum "asplain";
                enum "dot";
              }
            }
            leaf enable-client-to-client-reflection {
              description
                "Manages client to client route reflection.";
              type boolean;
              default "true";
            }
            leaf cluster-id {
              description
                "Configure Route-Reflector Cluster-id.";
```

```
              type string;
            }
            container confederation {
              description
                "AS confederation parameters.";
              leaf identifier {
                description
                  "Confederation identifier.";
                type string;
              }
              list peers {
                description
                  "Confederation peers.";
                key "as-name";
                leaf as-name {
                  type string;
                }
              }
            }
            container consistency-checker {
              description
                "Consistency-checker configuration.";
              leaf enable {
                type boolean;
              }
              leaf interval {
                description
                  "Check interval in minutes.";
                type uint16 {
                  range "5..1440";
                }
              }
              choice inconsistency-action {
                case error-message {
                  description
                    "Specifies that when an inconsistency is found, the system w
ill only generate a syslog message.";
                  leaf generate-error-message-only {
                    type boolean;
                  }
                }
                case autorepair {
                  description
                    "Specifies that when an inconsistency is found,
                     the system will generate a syslog message and take action
                     based on the type of inconsistency found.";
                  leaf perform-autorepair {
                    type boolean;
                  }
```

```
                }
              }
            }
            leaf deterministic-med {
              description
                "If enabled it enforce the deterministic comparison of the MED v
alue between
                 all paths received from within the same autonomous system.";
              type boolean;
            }
            container graceful-restart {
              description
                "Controls the BGP graceful restart capability.";
              leaf enable {
                type boolean;
              }
              leaf restart-time {
                description
                  "Sets the maximum time period (in seconds) that the local rout
er will wait
                   for a graceful-restart-capable neighbor to return to normal o
peration after a restart event occurs.";
                type uint16 {
                  range "1..3600";
                }
                default "120";
              }
              leaf stalepath-time {
                description
                  "Sets the maximum time period that the local router will hold
stale paths for a restarting peer.";
                type uint16 {
                  range "5..3600";
                }
                default "360";
              }
            }
            container listener-congfig {
              description
                "Associates a subnet range with a BGP peer group and activate th
e BGP dynamic neighbors feature.";
              leaf enable {
                type boolean;
              }
              leaf limit {
                description
                  "Sets a maximum limit number of BGP dynamic subnet range neigh
bors.";
                type uint16 {
                  range "1..5000";
                }
                default "100";
              }
              leaf range {
```

```
                description
                  "Specifies a subnet range that is to be associated with a spec
ified peer group.";
                type uint16 {
                  range "0..32";
                }
              }
              leaf peer-group {
                description
                  "Specifies a BGP peer group that is to be associated with the
specified subnet range.";
                type string;
              }
            }
            leaf log-neighbor-changes {
              description
                "Log neighbor up/down and reset reason.";
              type boolean;
            }
            leaf max-as-limit {
              description
                "Configures BGP to discard routes that have a number of autonomo
us system numbers in AS-path that exceed the specified value.";
              type uint16 {
                range "1..254";
              }
            }
            container router-id {
              description
                "Configures a fixed router ID for the local BGP routing process.
";
              leaf enable {
                type boolean;
              }
              choice config-type {
                case static {
                  leaf ip-address {
                    type boolean;
                  }
                }
                case auto-config {
                  leaf enable-auto-config {
                    type boolean;
                  }
                }
              }
            }
            container transport {
              description
                "Manages transport session parameters.";
              leaf enable-path-mtu-discovery {
                description
```

```
                  "Enables transport path MTU discovery.";
                type boolean;
                default "true";
              }
            }
          }
          leaf auto-summary {
            description
              "Enable automatic network number summarization";
            type boolean;
          }
          uses router-af-config;
          uses maximum-paths;
          leaf synchronization {
            description
              "Perform IGP synchronization.";
            type boolean;
          }
        }
        container mvpn {
          container bgp {
            description
              "BGP specific commands for ipv4-mvpn address family/sub-addess fam
ily combination.";
            uses bgp-af-mvpn-config;
          }
          leaf auto-summary {
            description
              "Enable automatic network number summarization.";
            type boolean;
          }
        }
      }
      container ipv6 {
        container multicast {
          container bgp {
            description
              "BGP specific commands for ipv6-multicast address family/sub-addes
s family combination.";
            uses bgp-af-config;
          }
          uses router-af-config;
        }
        container unicast {
          container bgp {
            description
              "BGP specific commands for ipv6-unicast address family/sub-address
family combination.";
            uses bgp-af-config;
          }
          uses router-af-config;
```

```
          leaf default-metric {
            description
              "Set metric of redistributed routes.";
            type uint32;
          }
          uses maximum-paths;
          leaf synchronization {
            description
              "Perform IGP synchronization.";
            type boolean;
          }
        }
        container mvpn {
          container bgp {
            description
              "BGP specific commands for ipv6-mvpn address family/sub-addess fam
ily combination.";
            uses bgp-af-mvpn-config;
          }
        }
      }
      container l2vpn {
        container vpls {
          container bgp {
            description
              "BGP specific commands for l2vpn-vpls address family/sub-addess fa
mily combination.";
            leaf scan-time {
              description
                "Configure background scanner interval in seconds.";
              type uint8 {
                range "5..60";
              }
            }
            uses slow-peer-config;
          }
        }
      }
      container nsap {
        container unicast {
          container bgp {
            description
              "BGP specific commands for nsap-unicast address family/sub-addess
family combination.";
            container aggregate-timer {
              description
                "Configure Aggregation Timer.";
              leaf enable {
                type boolean;
                default "true";
              }
```

```
              leaf threshold {
                type uint16 {
                  range "6..60";
                }
              }
            }
            leaf dampening {
              description
                "Enable route-flap dampening.";
              type boolean;
              default "false";
            }
            leaf propagate-dmzlink-bw {
              description
                "Use DMZ Link Bandwidth as weight for BGP multipaths.";
              type boolean;
            }
            leaf redistribute-internal {
              description
                "Allow redistribution of iBGP into IGPs (dangerous)";
              type boolean;
            }
            leaf scan-time {
              description
                "Configure background scanner interval in seconds.";
              type uint8 {
                range "5..60";
              }
            }
            uses slow-peer-config;
            leaf soft-reconfig-backup {
              description
                "Use soft-reconfiguration inbound only when route-refresh is not
 negotiated.";
              type boolean;
            }
          }
          leaf default-metric {
            description
              "Set metric of redistributed routes.";
            type uint32;
          }
          uses maximum-paths;
          leaf network {
            description
              "Specify a network to announce via BGP.";
            type inet:ip-address;
          }
          uses redistribute;
```

```
          leaf synchronization {
            description
              "Perform IGP synchronization.";
            type boolean;
          }
        }
      }
    }
    container rtfilter {
      container unicast {
        container bgp {
          description
            "BGP specific commands for rtfilter-unicast address family/sub-addes
s family combination.";
          uses slow-peer-config;
        }
        uses maximum-paths;
      }
    }
    container vpnv4 {
      container unicast {
        container bgp {
          description
            "BGP specific commands for vpnv4-unicast address family/sub-addess f
amily combination.";
          uses bgp-af-vpn-config;
        }
        uses maximum-paths;
      }
      container multicast {
        container bgp {
          description
            "BGP specific commands for vpnv4-multicast address family/sub-addess
 family combination.";
          uses bgp-af-vpn-config;
        }
        uses maximum-paths;
      }
    }
    container vpnv6 {
      container unicast {
        container bgp {
          description
            "BGP specific commands for vpnv6-unicast address family/sub-addess f
amily combination.";
          uses bgp-af-vpn-config;
        }
      }
    }
  }
  container bgp-neighbors {
    description
```

```
      "The top level container for the list of neighbours of the BGP router.";
    list bgp-neighbor {
      key "as-number";
      leaf as-number {
        type uint32;
      }
      choice peer-address-type {
        case ip-address {
          leaf ip-address {
            type inet:ip-address;
            mandatory true;
          }
        }
        case prefix {
          leaf prefix {
            type inet:ip-prefix;
            mandatory true;
          }
        }
        case host {
          leaf ip-host-address {
            type inet:host;
            mandatory true;
          }
        }
      }
      leaf prefix-list {
        type prefix-list-ref;
      }
      leaf default-action {
        type actions-enum;
      }
      container af-specific-config {
        description
          "Address family specific configuration parameters for the neighbours."
;
        container ipv4 {
          container mdt {
            uses neighbour-common-af-config;
          }
          container unicast {
            uses neighbour-ip-unicast-af-config;
          }
          container multicast {
            uses neighbour-ip-multicast-af-config;
          }
          container mvpn {
            uses neighbour-cast-af-config;
          }
```

```
        }
        container ipv6 {
          container unicast {
            uses neighbour-ip-unicast-af-config;
          }
          container multicast {
            uses neighbour-ip-multicast-af-config;
          }
          container mvpn {
            uses neighbour-common-af-config;
          }
        }
        container l2vpn {
          container evpn {
            uses neighbour-common-af-config;
          }
          container vpls {
            uses neighbour-common-af-config;
          }
        }
        container nsap {
          container unicast {
            uses neighbour-base-af-config;
            leaf prefix-list {
              type prefix-list-ref;
            }
          }
        }
        container rtfilter {
          container unicast {
            uses neighbour-base-af-config;
            leaf soft-reconfiguration {
              description
                "Allow inbound soft reconfiguration.";
              type boolean;
            }
          }
        }
        container vpnv4 {
          container unicast {
            uses neighbour-cast-af-config;
          }
          container multicast {
            uses neighbour-cast-af-config;
          }
        }
        container vpnv6 {
          container unicast {
```

```
            uses neighbour-cast-af-config;
          }
          container multicast {
            uses neighbour-cast-af-config;
          }
        }
      }
      container bgp-neighbor-state {
        description
          "The operational parameters describing the neighbour state.
           It is intended that this container may be augmented by vendors to ref
lect the vendor-specific operational state parameters.";
        leaf adminStatus {
          type bgp-peer-admin-status;
        }
        leaf in-lastupdatetime {
          type yang:timestamp;
        }
      }
      container bgp-neighbor-statistics {
        description
          "The operational parameters describing the neighbour statistics.
           It is intended that this container may be augmented by vendors to ref
lect the vendor-specific staistical parameters.";
        leaf nr-in-updates {
          type uint32;
        }
        leaf nr-out-updates {
          type uint32;
        }
      }
    }
  }
  container prefix-lists {
    description
      "Contains all prefix lists defined
       on a router.";
    list prefix-list {
      key "prefix-list-name";
      description
        "A prefix list.";
      leaf prefix-list-name {
        type string;
      }
      container prefixes {
        list prefix {
          key "seq-nr";
          description
            "A prefix is a rule with a BGP filter.
             The left hand side of the rule is the prefix filter.
```

```
             It specifies a set of IP addresses.
             If a BGP announcement contains an address that matches, the
             rule is applied.  The right hand side of the rule specifies
             the action that is to be applied.";
           leaf seq-nr {
             type uint16;
             description
               "Sequence number of the rule.
                The sequence number is included for compatibility purposes
                with CLI; from a machine-to-machine interface perspective,
                it would strictly speaking not be required as list elements
                can be arranged in a particular order.";
           }
           container prefix-filter {
             choice ip-address-group {
               case ip-address {
                 leaf ip-address {
                   type inet:ip-address;
                   mandatory true;
                 }
               }
               case prefix {
                 leaf prefix {
                   type inet:ip-prefix;
                   mandatory true;
                 }
               }
               case host {
                 leaf ip-host-address {
                   type inet:host;
                   mandatory true;
                 }
               }
               case ip-range {
                 leaf lower {
                   type inet:ip-address;
                 }
                 leaf upper {
                   type inet:ip-address;
                 }
               }
             }
             leaf action {
               type actions-enum;
               mandatory true;
               description
                 "permit/deny action";
             }
```

```
            container statistics {
              leaf prefix-hit-count {
                type uint32;
                config false;
              }
            }
          }
        }
      }
    }
  }
}
</CODE ENDS>
```

5. IANA Considerations

6. Security Considerations

The transport protocol used for sending the BGP data MUST support authentication and SHOULD support encryption. The data-model by itself does not create any security implications.

This draft does not change any underlying security issues inherent in [I-D.ietf-netmod-routing-cfg].

7. Acknowledgements

The authors would like to thank the reviewers of this document for their comments.

8. References

8.1. Normative References

[I-D.ietf-netmod-routing-cfg]
Lhotka, L., "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-15 (work in progress), May 2014.

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

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC2842] Chandra, R. and J. Scudder, "Capabilities Advertisement with BGP-4", RFC 2842, May 2000.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

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

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

8.2. Informative References

[RFC5492] Scudder, J. and R. Chandra, "Capabilities Advertisement with BGP-4", RFC 5492, February 2009.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, May 2014.

Authors' Addresses

Aleksandr Zhdankin
Cisco
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: azhdanki@cisco.com

Keyur Patel
Cisco
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: keyupate@cisco.com

Alexander Clemm
Cisco
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: alex@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 22, 2015

S. Zhuang
Z. Li
Sam aldrin
Huawei Technologies
J. Tantsura
G. Mirsky
Ericsson

August 21, 2014

BGP Link-State Extensions for Seamless BFD
draft-zhuang-idr-bgp-ls-sbfd-extensions-00

Abstract

[I-D.ietf-bfd-seamless-base] defines a simplified mechanism to use Bidirectional Forwarding Detection (BFD) with large portions of negotiation aspects eliminated, thus providing benefits such as quick provisioning as well as improved control and flexibility to network nodes initiating the path monitoring. The link-state routing protocols (IS-IS, OSPF and OSPFv3) have been extended to advertise the Seamless BFD (S-BFD) Discriminators.

This draft defines extensions to the BGP Link-state address-family to carry the S-BFD Discriminators information via BGP.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on February 22, 2015.

Table of Contents

1. Introduction

[I-D.ietf-bfd-seamless-base] defines a simplified mechanism to use Bidirectional Forwarding Detection (BFD) with large portions of negotiation aspects eliminated, thus providing benefits such as quick provisioning as well as improved control and flexibility to network nodes initiating the path monitoring.

[I-D.ginsberg-isis-sbfd-discriminator] defines a mean of advertising one or more S-BFD Discriminators using the IS-IS Router Capability TLV. [I-D.bhatia-ospf-sbfd-discriminator] defines a new OSPF Router Information (RI) TLV that allows OSPF routers to flood the S-BFD discriminator values associated with a target network identifier. This mechanism is applicable to both OSPFv2 and OSPFv3.

The link-state routing protocols (IS-IS, OSPF and OSPFv3) have been extended to advertise the S-BFD Discriminators. But flooding based propagation of the S-BFD Discriminators using IGPs is limited by the

perimeter of the IGP domain. For advertising the S-BFD Discriminators which span across IGP domains (e.g. multiple ASes), the Border Gateway Protocol (BGP) is better suited as its propagation perimeter is not limited like the IGPs.

This draft defines extensions to the BGP Link-state address-family to carry the S-BFD Discriminators information via BGP.

2. Terminology

This memo makes use of the terms defined in [I-D.ietf-bfd-seamless-base].

3. Problem and Requirement

Seamless MPLS [I-D.ietf-mpls-seamless-mpls] extends the core domain and integrates aggregation and access domains into a single MPLS domain. In a large network, the core and aggregation networks can be organized as different autonomous systems. Although the core and aggregation networks are segmented into different autonomous systems, but an E2E LSP will be created using hierarchical-labeled BGP LSPs based on iBGP-labeled unicast within each AS, and eBGP-labeled unicast to extend the LSP across AS boundaries. Meanwhile, the customer will see only two service-end points in the Seamless MPLS network. In order to detect the possible failure quickly and protect the network/trigger re-routing, BFD MAY be used for the Service Layer (e.g. for MPLS VPNs, PW) and the Transport Layer, so the need arises that the BFD session has to span across AS domain.

The link-state routing protocols (IS-IS, OSPF and OSPFv3) have been extended to advertise the S-BFD Discriminators. But flooding based propagation of the S-BFD Discriminators using IGPs is limited by the perimeter of the IGP domain. For advertising the S-BFD Discriminators which span across IGP domains (e.g. multiple ASes), the Border Gateway Protocol (BGP) is better suited as its propagation perimeter is not limited like the IGPs. This draft defines extensions requirement to the BGP Link-state address-family to carry the S-BFD Discriminators information via BGP.

4. BGP-LS Extensions for S-BFD Discriminators Exchanging

The BGP-LS NLRI can be a node NLRI, a link NLRI or a prefix NLRI. The corresponding BGP-LS attribute is a node attribute, a link attribute or a prefix attribute. BGP-LS [I-D.ietf-idr-ls-distribution] defines the TLVs that map link-state information to BGP-LS NLRI and BGP-LS attribute. This document adds additional BGP-LS attribute TLVs to encode the S-BFD Discriminators information.

[I-D.ginsberg-isis-sbfd-discriminator] defines the following TLVs to encode the S-BFD Discriminators information.

The ISIS Router CAPABILITY TLV as defined in [RFC4971] will be used to advertise S-BFD discriminators. A new Sub-TLV is defined as described below. S-BFD Discriminators Sub-TLV is formatted as specified in [RFC5305].

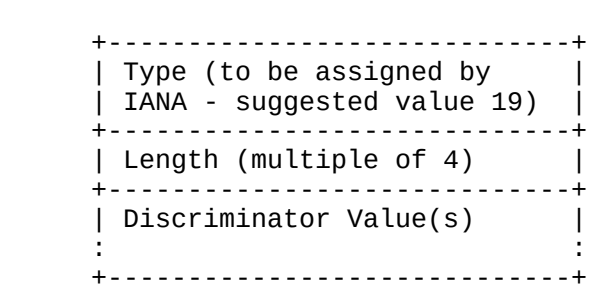


Figure 1: S-BFD Discriminators Sub-TLV

[I-D.bhatia-ospf-sbfd-discriminator] defines the following TLVs to encode the S-BFD Discriminators information. The format of the S-BFD Discriminator TLV is as follows:

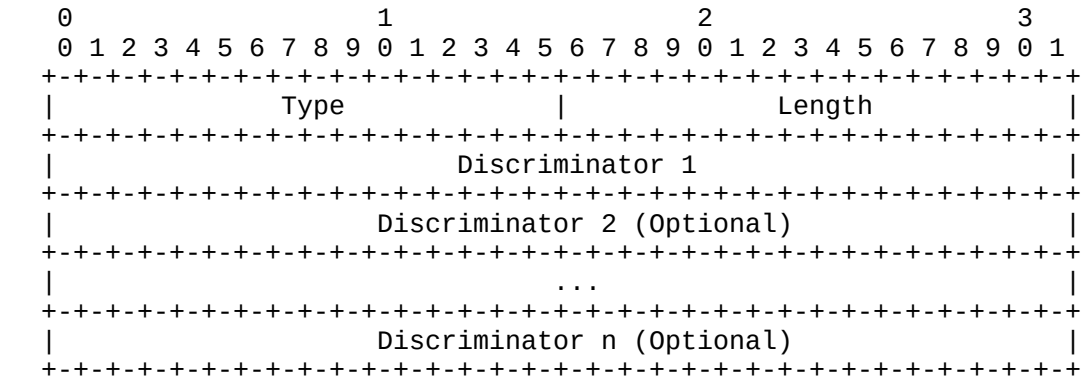


Figure 2: S-BFD Discriminators Sub-TLV

Type - S-BFD Discriminator TLV Type

Length - Total length of the discriminator (Value field) in octets, not including the optional padding. The Length is a multiple of 4 octets, and consequently specifies how many Discriminators are included in the TLV.

Value - S-BFD network target discriminator value or values.

Routers that do not recognize the S-BFD Discriminator TLV Type MUST ignore the TLV. S-BFD discriminator is associated with the BFD Target Identifier type, which allows de-multiplexing to a specific task or service.

These TLVs are mapped to BGP-LS attribute TLVs in the following way. The new information in the Link-State NLRIs and attributes is encoded in Type/Length/Value triplets.

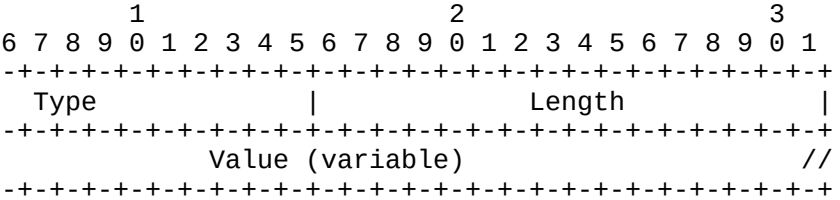


Figure 3: BGP-LS TLV format

The 2 octet Type field values are defined in Table 1. The next 2 octet Length field encodes length of the rest of the TLV. The Value portion of the TLV is variable and is equal to the corresponding Value portion of the TLV defined in [I-D.ginsberg-isis-sbfd-discriminator] and [I-D.bhatia-ospf-sbfd-discriminator].

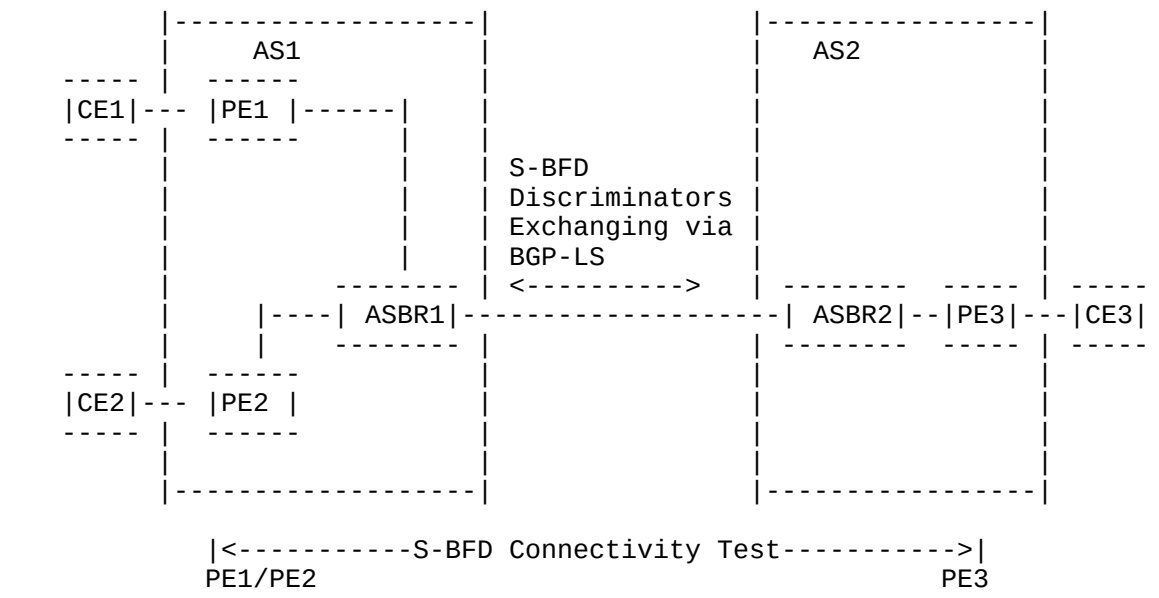
The following 'Node Attribute' TLVs are defined:

TLV Code Point	Description	Length	ISIS/OSPF TLV/Sub-TLV
TBD	S-BFD Discriminators	variable	TBD
...	...	...	...

Table 1: Node Attribute TLVs

5. Operation

In an inter-as VPN network as follows, ASBR1 and ASBR2 establish a BGP-LS session for exchanging S-BFD Discriminators information.



Step 1: ASBR1 learns all the S-BFD Discriminators information within AS1 by the mean defines in [I-D.ginsberg-isis-sbfd-discriminator] or [I-D.bhatia-ospf-sbfd-discriminator].

Step 2: ASBR1 sends AS1's S-BFD Discriminators information to AS2's ASBR2 via the BGP-LS session.

Step 3: ASBR2 injects the AS1's S-BFD Discriminators information receiving from ASBR1 into IGP (IS-IS or OSPF or OSPFv3), then flood them within the domain of the AS2 via IGP, So the nodes of AS2 can learn all the S-BFD Discriminators information originating from AS1.

Likewise, the nodes of AS1 can learn all the S-BFD Discriminators information originating from AS2.

At this point, we can use S-BFD Procedures defines in [I-D.ietf-bfd-seamless-base] between the PEs which belong to different AS.

6. IANA Considerations

TBD.

7. Security Considerations

This document does not introduce any new security risk.

8. Acknowledgements

The authors would like to thank Nan Wu for his contributions to this work.

9. References

9.1. Normative References

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

9.2. Informative References

[I-D.bhatia-ospf-sbfd-discriminator]
Bhatia, M., Ranganath, T., Pignataro, C., and S. Aldrin, "OSPF extensions to advertise S-BFD Target Discriminator", draft-bhatia-ospf-sbfd-discriminator-00 (work in progress), May 2014.

[I-D.ginsberg-isis-sbfd-discriminator]
Ginsberg, L., Akiya, N., and M. Chen, "Advertising S-BFD Discriminators in IS-IS", draft-ginsberg-isis-sbfd-discriminator-00 (work in progress), May 2014.

[I-D.ietf-bfd-seamless-base]
Akiya, N., Pignataro, C., Ward, D., Bhatia, M., and J. Networks, "Seamless Bidirectional Forwarding Detection (S-BFD)", draft-ietf-bfd-seamless-base-02 (work in progress), August 2014.

[I-D.ietf-idr-ls-distribution]
Gredler, H., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and TE Information using BGP", draft-ietf-idr-ls-distribution-05 (work in progress), May 2014.

[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.

[RFC4971] Vasseur, JP., Shen, N., and R. Aggarwal, "Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information", RFC 4971, July 2007.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.

Authors' Addresses

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

Email: zhuangshunwan@huawei.com

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

Email: lizhenbin@huawei.com

Sam Aldrin
Huawei Technologies
2330 Central Expressway
Santa Clara CA 95051

Email: sam.aldrin@huawei.com

Jeff Tantsura
Ericsson
200 Holger Way
San Jose CA 95134
USA

Email: jeff.tantsura@ericsson.com

Greg Mirsky
Ericsson
300 Holger Way
San Jose CA 95134
USA

Email: gregory.mirsky@ericsson.com