Network Working Group Yakov Rekhter (Juniper Networks)
Internet Draft Rahul Aggarwal (Redback Networks)
Expiration Date: July 2002

Graceful Restart Mechanism for BGP with MPLS

draft-ietf-mpls-bgp-mpls-restart-00.txt

1. Status of this Memo

2. Abstract

[1] describes a mechanism for BGP that would help minimize the negative effects on routing caused by BGP restart. This document extends this mechanism to also minimize the negative effects on MPLS forwarding when BGP is used to carry MPLS labels [2]. The mechanism described in this document is agnostic with respect to the types of the addresses carried in the BGP NLRI. As such it works in conjunction with any of the address famililies that could be carried in BGP (e.g., IPv4, IPv6, etc...)

3. Summary for Sub-IP Area

3.1. Summary

This document describes a mechanism that helps to minimize the negative effects on MPLS forwarding caused by LSR's control plane restart, and specifically by the restart of its BGP component in the case where BGP is used to carry MPLS labels and LSR is capable of preserving its MPLS forwarding state across the restart.

3.2. Related documents

See the Reference Section

3.3. Where does it fit in the Picture of the Sub-IP Work

This work fits squarely in MPLS box.

3.4. Why is it Targeted at this WG

The specifications on carrying MPLS Labels in BGP is a product of the MPLS WG. This document specifies procedures to minimize the negative effects on MPLS forwarding caused by the restart of the control plane BGP module in the case where BGP is used to carry MPLS labels. Since the procedures described in this document are directly related to MPLS forwarding and carrying MPLS labels in BGP, it would be logical to target this document at the MPLS WG.

3.5. Justification

The WG should consider this document, as it allows to minimize the negative effects on MPLS forwarding caused by the restart of the control plane BGP module in the case where BGP is used to carry MPLS labels.

4. Motivation

In the case where an LSR could preserve its MPLS forwarding state across restart of its control plane, and specifically its BGP component, it may be desirable not to perturb the LSPs going through that LSR (and specifically, the LSPs established by BGP). In this document, we describe a mechanism that allows to accomplish this goal. The mechanism described in this document works in conjunction with the mechanism specified in [1]. The mechanism described in this document places no restrictions on the types of addresses (address families) that it can support.

5. Assumptions

First of all we assume that an LSR implements the Graceful Restart Mechanism for BGP, as specified in [1]. Second, we assume that the LSR is capable of preserving its MPLS forwarding state across the restart of its control plane (including the restart of BGP).

6. Capability Advertisement

An LSR that supports the mechanism described in this document advertises this to its peer by using the Graceful Restart Capability, as specified in [1]. The SAFI in the advertised capability should indicate that NLRI carries not just address prefixes but labels as well.

7. Procedures for the restarting LSR

After the LSR restarts, it follows the procedures as specified in [1]. In addition, if the LSR is able to preserve its MPLS forwarding state across the restart, the LSR advertises this to its neighbors by appropriately setting the Flag field in the Graceful Restart Capability for all applicable AFI/SAFI pairs.

For the sake of brevity in the context of this document by "MPLS forwarding state" we mean either <incoming label -> (outgoing label, next hop)>, or <address prefix -> (outgoing label, next hop)> mapping. In the context of this document the forwarding state that is referred to in [1] means MPLS forwarding state. This document doesn't require the restarting LSR to preserve its IP forwarding state across the restart.

Once the restarting LSR completes its route selection (as specified in Section 6.1 of [1]), then in addition to the procedures specified

in [1], the restarting LSR performs one of the following:

7.1. Case 1

The following applies when (a) the best route selected by the restarting LSR was received with a label, (b) that label is not an Implicit NULL, and (c) the LSR advertises this route with itself as the next hop.

In this case the restarting LSR searches its MPLS forwarding state (the one preserved across the restart) for an entry with <outgoing label, Next-Hop> equal to the one in the received route. If such an entry is found, the LSR no longer marks the entry as stale. In addition if the entry is of type <incoming label, (outgoing label, next hop)> rather than <prefix, (outgoing label, next hop)>, the LSR uses the incoming label from the entry when advertising the route to its neighbors. If the found entry has no incoming label, or if no such entry is found, the LSR just picks up some unused label when advertising the route to its neighbors (assuming that there are neighbors to which the LSR has to advertise the route with a label).

7.2. Case 2

The following applies when (a) the best route selected by the restarting LSR was received either without a label, or with an Implicit NULL label, or the route is originated by the restarting LSR, (b) the LSR advertises this route with itself as the next hop, and (c) the LSR has to generate a (non Implicit NULL) label for the route.

In this case the LSR searches its MPLS forwarding state for an entry that indicates that the LSR has to perform label pop, and the next hop equal to the next hop of the route in consideration. If such an entry is found, then the LSR uses the incoming label from the entry when advertising the route to its neighbors. If no such entry is found, the LSR just picks up some unused label when advertising the route to its neighbors.

The description in the above paragraph assumes that the restarting LSR generates the same label for all the routes with the same next hop. If this is not the case, and the restarting LSR generates a unique label per each such route, then the LSR needs to preserve across the restart not just <incoming label, (outgoing label, next hop)> mapping, but also the prefix associated with this mapping. In such case the LSR would search its MPLS forwarding state for an entry that (a) indicates Label pop (means no outgoing label), (b) the next

hop equal to the next hop of the route and (c) has the same prefix as the route. If such an entry is found, then the LSR uses the incoming label from the entry when advertising the route to its neighbors. If no such entry is found, the LSR just picks up some unused label when advertising the route to its neighbors.

7.3. Case 3

The following applies when the restarting LSR does not set BGP Next Hop to self.

In this case the restarting LSR, when advertising its best route for a particular NLRI just uses the label that was received with that route. And if the route was received with no label, the LSR advertises the route with no label as well.

8. Alternative procedures for the restarting LSR

In this section we describe an alternative to the procedures described in Section 7.

The procedures described in this section assume that the restarting LSR has (at least) as many unallocated as allocated labels. The latter forms the MPLS forwarding state that the LSR managed to preserve across the restart. The former is used for allocating labels after the restart.

After the LSR restarts, it follows the procedures as specified in [1]. In addition, if the LSR is able to preserve its MPLS forwarding state across the restart, the LSR advertises this to its neighbors by appropriately setting the Flag field in the Graceful Restart Capability.

To create local label bindings the LSR uses unallocated labels (this is pretty much the normal procedure). That means that as long as the LSR retains the MPLS forwarding state that the LSR preserved across the restart, the labels from that state are not used for creating local label bindings.

The restarting LSR should retain the MPLS forwarding state that the LSR preserved across the restart at least until the LSR sends End-of-RIB marker to all of its neighbors (by that time the LSR already completed its route selection process, and also advertised its Adj-RIB-Out to its neighbors). It may be desirable to retain the forwarding state even a bit longer, as to allow the neighbors to receive and process the routes that have been advertised by the

restarting LSR. After that, the restarting LSR may delete the MPLS forwarding state that it preserved across the restart.

Note that while an LSR is in the process of restarting, the LSR may have not one, but two local label bindings for a given BGP route - one that was retained from prior to restart, and another that was created after the restart. Once the LSR completes its restart, the former will be deleted. Both of these bindings though would have the same outgoing label (and the same next hop).

9. Procedures for a neighbor of a restarting LSR

The neighbor of a restarting LSR (the receiving router in terminology used in [1]) follows the procedures specified in [1]. In addition, the neighbor should treat the MPLS labels received from the restarting LSR the same way as it treats the routes received from the restarting LSR (both prior and after the restart).

Replacing the stale routes by the routing updates received from the restarting LSR involves replacing/updating the appropriate MPLS labels.

In addition, if the Flags in the Graceful Restart Capability received from the restarting LSR indicate that the LSR wasn't able to retain its MPLS state across the restart, the neighbor should immediately remove all the NLRI and the associated MPLS labels that it previously acquired via BGP from the restarting LSR.

An LSR, once it creates a <label, FEC> binding, should keep the value of the label in this binding for as long as the LSR has a route to the FEC in the binding. If the route to the FEC disappears, and then re-appears again later, then this may result in using a different label value, as when the route re-appears, the LSR would create a new <label, FEC> binding.

To minimize the potential mis-routing caused by the label change, when creating a new <label, FEC> binding the LSR should pick up the least recently used label. Once an LSR releases a label, the LSR should not re-use this label for advertising a <label, FEC> binding to a neighbor that supports graceful restart for at least the Restart Time, as advertised by the neighbor to the LSR.

10. Security Consideration

This document does not introduce new security issues. The security considerations pertaining to the original BGP protocol remain relevant.

11. Intellectual Property Considerations

Juniper Networks, Inc. is seeking patent protection on some or all of the technology described in this Internet-Draft. If technology in this document is adopted as a standard, Juniper Networks agrees to license, on reasonable and non-discriminatory terms, any patent rights it obtains covering such technology to the extent necessary to comply with the standard.

Redback Networks, Inc. is seeking patent protection on some of the technology described in this Internet-Draft. If technology in this document is adopted as a standard, Redback Networks agrees to license, on reasonable and non-discriminatory terms, any patent rights it obtains covering such technology to the extent necessary to comply with the standard.

12. Acknowledgments

We would like to thank Chaitanya Kodeboyina for his review and comments. The approach described in Section 8 is based on the idea suggested by Manoj Leelanivas.

13. References

[1] "Graceful Restart Mechanism for BGP", draft-ietf-idr-restart-01.txt

[2] "Carrying Label Information in BGP-4", RFC3107

14. Author Information

Yakov Rekhter
Juniper Networks
1194 N.Mathilda Ave
Sunnyvale, CA 94089
e-mail: yakov@juniper.net

Rahul Aggarwal
Redback Networks
350 Holger Way
San Jose, CA 95134
e-mail: rahul@redback.com