Network Working Group
Request for Comments: 3478
Category: Standards Track

M. Leelanivas
Y. Rekhter
Juniper Networks
R. Aggarwal
Redback Networks

February 2003

Graceful Restart Mechanism for Label Distribution Protocol

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

Abstract

This document describes a mechanism that helps to minimize the negative effects on MPLS traffic caused by Label Switching Router's (LSR's) control plane restart, specifically by the restart of its Label Distribution Protocol (LDP) component, on LSRs that are capable of preserving the MPLS forwarding component across the restart.

The mechanism described in this document is applicable to all LSRs, both those with the ability to preserve forwarding state during LDP restart and those without (although the latter needs to implement only a subset of the mechanism described in this document). Supporting (a subset of) the mechanism described here by the LSRs that can not preserve their MPLS forwarding state across the restart would not reduce the negative impact on MPLS traffic caused by their control plane restart, but it would minimize the impact if their neighbor(s) are capable of preserving the forwarding state across the restart of their control plane and implement the mechanism described here.

The mechanism makes minimalistic assumptions on what has to be preserved across restart - the mechanism assumes that only the actual MPLS forwarding state has to be preserved; the mechanism does not require any of the LDP-related states to be preserved across the restart.

The procedures described in this document apply to downstream unsolicited label distribution. Extending these procedures to downstream on demand label distribution is for further study.

Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

1. Motivation

For the sake of brevity in the context of this document, by "the control plane" we mean "the LDP component of the control plane".

For the sake of brevity in the context of this document, by "MPLS forwarding state" we mean either <incoming label -> (outgoing label, next hop)> (non-ingress case), or <FEC->(outgoing label, next hop)> (ingress case) mapping.

In the case where a Label Switching Router (LSR) could preserve its MPLS forwarding state across restart of its control plane, specifically its LDP component [LDP], it is desirable not to perturb the LSPs going through that LSR (specifically, the LSPs established by LDP). In this document, we describe a mechanism, termed "LDP Graceful Restart", that allows the accomplishment of this goal.

The mechanism described in this document is applicable to all LSRs, both those with the ability to preserve forwarding state during LDP restart and those without (although the latter need to implement only a subset of the mechanism described in this document). Supporting (a subset of) the mechanism described here by the LSRs that can not preserve their MPLS forwarding state across the restart would not reduce the negative impact on MPLS traffic caused by their control plane restart, but it would minimize the impact if their neighbor(s) are capable of preserving the forwarding state across the restart of their control plane and implement the mechanism described here.

The mechanism makes minimalistic assumptions on what has to be preserved across restart - the mechanism assumes that only the actual MPLS forwarding state has to be preserved. Clearly this is the minimum amount of state that has to be preserved across the restart in order not to perturb the LSPs traversing a restarting LSR. The mechanism does not require any of the LDP-related states to be preserved across the restart.

In the scenario where label binding on an LSR is created/maintained not just by the LDP component of the control plane, but by other protocol components as well (e.g., BGP, RSVP-TE), and the LSR supports restart of the individual components of the control plane that create/maintain label binding (e.g., restart of LDP, but no restart of BGP), the LSR needs to preserve across the restart the information about which protocol has assigned which labels.

The procedures described in this document apply to downstream unsolicited label distribution. Extending these procedures to downstream on demand label distribution is for further study.

2. LDP Extension

An LSR indicates that it is capable of supporting LDP Graceful Restart, as defined in this document, by including the Fault Tolerant (FT) Session TLV as an Optional Parameter in the LDP Initialization message. The format of the FT Session TLV is defined in [FT-LDP]. The L (Learn from Network) flag MUST be set to 1, which indicates that the procedures in this document are used. The rest of the FT flags are set to 0 by a sender and ignored on receipt.

The value field of the FT Session TLV contains two components that are used by the mechanisms defined in this document: FT Reconnect Timeout, and Recovery Time.

The FT Reconnect Timeout is the time (in milliseconds) that the sender of the TLV would like the receiver of that TLV to wait after the receiver detects the failure of LDP communication with the sender. While waiting, the receiver SHOULD retain the MPLS forwarding state for the (already established) LSPs that traverse a link between the sender and the receiver. The FT Reconnect Timeout should be long enough to allow the restart of the control plane of the sender of the TLV, and specifically its LDP component to bring it to the state where the sender could exchange LDP messages with its neighbors.

Setting the FT Reconnect Timeout to 0 indicates that the sender of the TLV will not preserve its forwarding state across the restart, yet the sender supports the procedures, defined in Section 3.3, "Restart of LDP communication with a neighbor LSR" of this document, and therefore could take advantage if its neighbor to preserve its forwarding state across the restart.

For a restarting LSR, the Recovery Time carries the time (in milliseconds) the LSR is willing to retain its MPLS forwarding state that it preserved across the restart. The time is from the moment the LSR sends the Initialization message that carries the FT Session

TLV after restart. Setting this time to 0 indicates that the MPLS forwarding state was not preserved across the restart (or even if it was preserved, is no longer available).

The Recovery Time SHOULD be long enough to allow the neighboring LSR's to re-sync all the LSP's in a graceful manner, without creating congestion in the LDP control plane.

3. Operations

An LSR that supports functionality described in this document advertises this to its LDP neighbors by carrying the FT Session TLV in the LDP Initialization message.

This document assumes that in certain situations, as specified in section 3.1.2, "Egress LSR", in addition to the MPLS forwarding state, an LSR can also preserve its IP forwarding state across the restart. Procedures for preserving an IP forwarding state across the restart are defined in [OSPF-RESTART], [ISIS-RESTART], and [BGP-RESTART].

3.1. Procedures for the restarting LSR

After an LSR restarts its control plane, the LSR MUST check whether it was able to preserve its MPLS forwarding state from prior to the restart. If not, then the LSR sets the Recovery Time to 0 in the FT Session TLV the LSR sends to its neighbors.

If the forwarding state has been preserved, then the LSR starts its internal timer, called MPLS Forwarding State Holding timer (the value of that timer SHOULD be configurable), and marks all the MPLS forwarding state entries as "stale". At the expiration of the timer, all the entries still marked as stale SHOULD be deleted. The value of the Recovery Time advertised in the FT Session TLV is set to the (current) value of the timer at the point in which the Initialization message carrying the FT Session TLV is sent.

We say that an LSR is in the process of restarting when the MPLS Forwarding State Holding timer is not expired. Once the timer expires, we say that the LSR completed its restart.

The following procedures apply when an LSR is in the process of restarting.

3.1.1. Non-egress LSR

If the label carried in the newly received Mapping message is not an Implicit NULL, the LSR searches its MPLS forwarding state for an

entry with the outgoing label equal to the label carried in the message, and the next hop equal to one of the addresses (next hops) received in the Address message from the peer. 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 <FEC, (outgoing label, next hop)>), the LSR associates the incoming label from that entry with the FEC received in the Label Mapping message, and advertises (via LDP) <incoming label, FEC> to its neighbors. If the found entry has no incoming label, or if no entry is found, the LSR follows the normal LDP procedures. (Note that this paragraph describes the scenario where the restarting LSR is neither the egress, nor the penultimate hop that uses penultimate hop popping for a particular LSP. Note also that this paragraph covers the case where the restarting LSR is the ingress.)

If the label carried in the Mapping message is an Implicit NULL label, the LSR searches its MPLS forwarding state for an entry that indicates Label pop (means no outgoing label), and the next hop equal to one of the addresses (next hops) received in the Address message from the peer. If such an entry is found, the LSR no longer marks the entry as stale, the LSR associates the incoming label from that entry with the FEC received in the Label Mapping message from the neighbor, and advertises (via LDP) <incoming label, FEC> to its neighbors. If the found entry has no incoming label, or if no entry is found, the LSR follows the normal LDP procedures. (Note that this paragraph describes the scenario where the restarting LSR is a penultimate hop for a particular LSP, and this LSP uses penultimate hop popping.)

The description in the above paragraph assumes that the restarting LSR generates the same label for all the LSPs that terminate on the same LSR (different from the restarting LSR), and for which the restarting LSR is a penultimate hop. If this is not the case, and the restarting LSR generates a unique label per each such LSP, then the LSR needs to preserve across the restart, not just the <incoming label, (outgoing label, next hop)> mapping, but also the FEC associated with this mapping. In such case, the LSR searches its MPLS forwarding state for an entry that (a) indicates Label pop (means no outgoing label), (b) indicates the next hop equal to one of the addresses (next hops) received in the Address message from the peer, and (c) has the same FEC as the one received in the Label Mapping message. If such an entry is found, the LSR no longer marks the entry as stale, the LSR associates the incoming label from that entry with the FEC received in the Label Mapping message from the neighbor, and advertises (via LDP) <incoming label, FEC> to its neighbors. If the found entry has no incoming label, or if no entry is found, the LSR follows the normal LDP procedures.

3.1.2. Egress LSR

If an LSR determines that it is an egress for a particular FEC, the LSR is configured to generate a non-NULL label for that FEC, and that the LSR is configured to generate the same (non-NULL) label for all the FECs that share the same next hop and for which the LSR is an egress, the LSR searches its MPLS forwarding state for an entry that indicates Label pop (means no outgoing label), and the next hop equal to the next hop for that FEC. (Determining the next hop for the FEC depends on the type of the FEC. For example, when the FEC is an IP address prefix, the next hop for that FEC is determined from the IP forwarding table.) If such an entry is found, the LSR no longer marks this entry as stale, the LSR associates the incoming label from that entry with the FEC, and advertises (via LDP) <incoming label, FEC> to its neighbors. If the found entry has no incoming label, or if no entry is found, the LSR follows the normal LDP procedures.

If an LSR determines that it is an egress for a particular FEC, the LSR is configured to generate a non-NULL label for that FEC, and that the LSR is configured to generate a unique label for each such FEC, then the LSR needs to preserve across the restart, not just the <incoming label, (outgoing label, next hop)> mapping, but also the FEC associated with this mapping. In such case, the LSR would search its MPLS forwarding state for an entry that indicates Label pop (means no outgoing label), and the next hop equal to the next hop for that FEC associated with the entry (Determining the next hop for the FEC depends on the type of the FEC. For example, when the FEC is an IP address prefix, the next hop for that FEC is determined from the IP forwarding table.) If such an entry is found, the LSR no longer marks this entry as stale, the LSR associates the incoming label from that entry with the FEC, and advertises (via LDP) <incoming label, FEC> to its neighbors. If the found entry has no incoming label, or if no entry is found, the LSR follows the normal LDP procedures.

If an LSR determines that it is an egress for a particular FEC, and the LSR is configured to generate a NULL (either Explicit or Implicit) label for that FEC, the LSR just advertises (via LDP) such label (together with the FEC) to its neighbors.

3.2. Alternative procedures for the restarting LSR

In this section we describe an alternative to the procedures described in Section 3.1, "Procedures for the restarting LSR".

The procedures described in this section assumes that the restarting LSR has (at least) as many unallocated as allocated labels. The latter form the MPLS forwarding state that the LSR managed to preserve across the restart.

After an LSR restarts its control plane, the LSR MUST check whether it was able to preserve its MPLS forwarding state from prior to the restart. If no, then the LSR sets the Recovery Time to 0 in the FT Session TLV the LSR sends to its neighbors.

If the forwarding state has been preserved, then the LSR starts its internal timer, called MPLS Forwarding State Holding timer (the value of that timer SHOULD be configurable), and marks all the MPLS forwarding state entries as "stale". At the expiration of the timer, all the entries still marked as stale SHOULD be deleted. The value of the Recovery Time advertised in the FT Session TLV is set to the (current) value of the timer at the point when the Initialization message carrying the FT Session TLV is sent.

We say that an LSR is in the process of restarting when the MPLS Forwarding State Holding timer is not expired. Once the timer expires, we say that the LSR completed its restart.

While an LSR is in the process of restarting, the LSR creates local label binding by following the normal LDP procedures.

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 FEC - 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).

3.3. Restart of LDP communication with a neighbor LSR

When an LSR detects that its LDP session with a neighbor went down, and the LSR knows that the neighbor is capable of preserving its MPLS forwarding state across the restart (as was indicated by the FT Session TLV in the Initialization message received from the neighbor), the LSR retains the label-FEC bindings received via that session (rather than discarding the bindings), but marks them as "stale".

After detecting that the LDP session with the neighbor went down, the LSR tries to re-establish LDP communication with the neighbor following the usual LDP procedures.

The amount of time the LSR keeps its stale label-FEC bindings is set to the lesser of the FT Reconnect Timeout, as was advertised by the neighbor, and a local timer, called the Neighbor Liveness Timer. If within that time the LSR still does not establish an LDP session with the neighbor, all the stale bindings SHOULD be deleted. The Neighbor

Liveness Timer is started when the LSR detects that its LDP session with the neighbor went down. The value of the Neighbor Liveness timer SHOULD be configurable.

If the LSR re-establishes an LDP session with the neighbor within the lesser of the FT Reconnect Timeout and the Neighbor Liveness Timer, and the LSR determines that the neighbor was not able to preserve its MPLS forwarding state, the LSR SHOULD immediately delete all the stale label-FEC bindings received from that neighbor. If the LSR determines that the neighbor was able to preserve its MPLS forwarding state (as was indicated by the non-zero Recovery Time advertised by the neighbor), the LSR SHOULD further keep the stale label-FEC bindings, received from the neighbor, for as long as the lesser of the Recovery Time advertised by the neighbor, and a local configurable value, called Maximum Recovery Time, allows.

The LSR SHOULD try to complete the exchange of its label mapping information with the neighbor within 1/2 of the Recovery Time, as specified in the FT Session TLV received from the neighbor.

The LSR handles the Label Mapping messages received from the neighbor by following the normal LDP procedures, except that (a) it treats the stale entries in its Label Information Base (LIB) as if these entries have been received over the (newly established) session, (b) if the label-FEC binding carried in the message is the same as the one that is present in the LIB, but is marked as stale, the LIB entry is no longer marked as stale, and (c) if for the FEC in the label-FEC binding carried in the message there is already a label-FEC binding in the LIB that is marked as stale, and the label in the LIB binding is different from the label carried in the message, the LSR just updates the LIB entry with the new label.

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, 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 sum of the FT Reconnect Timeout plus Recovery Time, as advertised by the neighbor to the LSR.

4. Security Consideration

The security considerations pertaining to the original LDP protocol [RFC3036] remain relevant.

In addition, LSRs that implement the mechanism described here are subject to to additional denial-of-service attacks as follows:

An intruder may impersonate an LDP peer in order to force a failure and reconnection of the TCP connection, but where the intruder sets the Recovery Time to 0 on reconnection. This forces all labels received from the peer to be released.

An intruder could intercept the traffic between LDP peers and override the setting of the Recovery Time to be set to 0. This forces all labels received from the peer to be released.

All of these attacks may be countered by use of an authentication scheme between LDP peers, such as the MD5-based scheme outlined in [LDP].

As with LDP, a security issue may exist if an LDP implementation continues to use labels after expiration of the session that first caused them to be used. This may arise if the upstream LSR detects the session failure after the downstream LSR has released and re-used the label. The problem is most obvious with the platform-wide label space and could result in mis-routing data to other than intended destinations, and it is conceivable that these behaviors may be deliberately exploited to either obtain services without authorization or to deny services to others.

In this document, the validity of the session may be extended by the Reconnect Timeout, and the session may be re-established in this period. After the expiry of the Reconnection Timeout, the session must be considered to have failed and the same security issue applies as described above.

However, the downstream LSR may declare the session as failed before the expiration of its Reconnection Timeout. This increases the period during which the downstream LSR might reallocate the label while the upstream LSR continues to transmit data using the old usage of the label. To reduce this issue, this document requires that labels not be re-used until at least the sum of Reconnect Timeout plus Recovery Time.

5. Intellectual Property Considerations

This section is taken from Section 10.4 of [RFC2026].

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

The IETF has been notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

6. Acknowledgments

We would like to thank Loa Andersson, Chaitanya Kodeboyina, Ina Minei, Nischal Sheth, Enke Chen, and Adrian Farrel for their contributions to this document.

7. Normative References

[LDP] Andersson, L., Doolan, P., Feldman, N., Fredette, A. and B. Thomas, "Label Distribution Protocol", RFC 3036, January 2001.

[FT-LDP] Farrel, A., "Fault Tolerance for the Label Distribution Protocol (LDP)", RFC 3479, February 2003.

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

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

8. Informative References

[OSPF-RESTART] "Hitless OSPF Restart", Work in Progress.

[ISIS-RESTART] "Restart signaling for ISIS", Work in Progress.

[BGP-RESTART] "Graceful Restart Mechanism for BGP", Work in Progress.

9. Authors' Addresses

Manoj Leelanivas
Juniper Networks
1194 N. Mathilda Ave
Sunnyvale, CA 94089

EMail: manoj@juniper.net

Yakov Rekhter
Juniper Networks
1194 N. Mathilda Ave
Sunnyvale, CA 94089

EMail: yakov@juniper.net

Rahul Aggarwal
Redback Networks
350 Holger Way
San Jose, CA 95134

EMail: rahul@redback.com

10. Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.