Network Working Group
Internet Draft
Expiration Date: December 2006

Bob Thomas
Cisco Systems, Inc.

June 2006

LDP Capabilities

draft-thomas-mpls-ldp-capabilities-00.txt

Status of this Memo

Copyright Notice

Abstract

A number of enhancements to LDP as specified in [RFC3036] have been proposed. Some have been implemented, and some are advancing toward standardization. It is likely that additional enhancements will be proposed in the future. At present LDP has no common mechanism for LDP speakers to activate such enhancements. Consequently, the enhancement specifications typically include ad hoc procedures for activating the enhancement at session initialization time. This

draft specifies an LDP mechanism for activating enhancements that allows LDP peers to enable and disable them at any time following session establishment.

1. Introduction

A number of enhancements to LDP as specified in [RFC3036] have been proposed. These include LDP Graceful Restart [RFC3478], Fault Tolerant LDP [RFC3479], multicast extensions [MLDP], signaling for level 2 circuits [PWE], learning labels from next nexthop routers for FRR [NNH], upstream label allocation [UPSTREAM], and extensions for signaling inter-area LSPs [IALDP]. Some of these have been implemented, and some are advancing toward standardization. It is likely that additional enhancements will be proposed in the future.

At present LDP has no common mechanism for LDP speakers to activate such enhancements. Consequently, the enhancement specifications typically include ad hoc procedures for activating the enhancement at session initialization time or assume that the enhancement is always active. Furthermore, LDP has no mechanism for de-activating an enhancement once activated.

This draft specifies an LDP capability mechanism for managing the use of enhancements that:

- Associates capabilities with LDP enhancements.

- Enables LDP peers to activate an enhancement at session establishment time or any time following session establishment by enabling the corresponding capability;

- Enables an LDP peer to deactivate an enhancement at any time by disabling the corresponding capability;

- Supports enhancements that are asymmetric as well as ones that are symmetric;

- Is "enhancement independent" in that the mechanism is general enough to support a wide range of enhancements.

With the mechanism in place an optional LDP capability would be specified by a document that:

- Describes the motivation for the enhancement.

- Specifies the behavior of LDP when the enhancement is enabled.

- Specifies that the LDP extension mechanism is to be used to enable the capability corresponding to the enhancement;

- To the extent that the extension has parameters, describes the method for encoding the parameters.

- Includes an IANA considerations section that notes that an IANA-assigned code point for the capability corresponding to the enhancement is required.

2. LDP Capability Mechanism

Enhancements are likely to be activated during LDP session establishment as each peer enables the capabilities corresponding to the enhancments it desires. Beyond that, however, capabilities can be used to dynamically modify the characteristics of the session to suit changing conditions. For example, an LSR capable of supporting a particular capability in support of some "feature" may not have enabled the capability with its peers at session establishment time because the feature was disabled at that time. Later an operator may enable the feature, at which time the LSR would react by announcing to its peers that the capability is enabled. Similarly, when a capability is in use in support of a feature and an operator disables that feature, the LSR reacts by announcing to its peers that the capability is disabled.

The expectation is that many LDP enhancements will require both session peers perform actions in support of the enhancement. Such an enhancement is said to be "symmetric"; each peer must enable the corresponding capability.

An enhancement may be "asymmetric" in that it requires that only one session peer perform actions in support of the enhancement. For example, RFC3036 explicitly prohibits use of the Wildcard FEC with Label Request messages. The capability for an enhancement that overrides this prohibition requires only the peer willing to process Label Request messages with the Wildcard FEC enable it. Once enabled the other peer may send such messages; until it is enabled the other peer may not.

The mechanism for enabling LDP capabilities is similar to the BGP dynamic capability mechanism [BGP_DYNAMIC].

The mechanism operates as follows:

1. Each LDP speaker is assumed to implement a set of capabilities. At any time a speaker may have none, one or more of those capabilities "enabled". When a capability is enabled the speaker shall perform protocol actions specified for that capability. For example, such actions may involve receiving and processing messages from a peer that the capability requires. Unless a capability is enabled the speaker will not perform protocol actions specified for that capability.

2. At session establishment time the LDP speaker announces its capabilities that are currently enabled in its Initialization message.

3. At any time after session establishment the LDP speaker may inform a peer that a capability not previously enabled is now enabled by sending a Capability message that specifies the capability. Likewise, the LDP speaker may inform a peer that one of its capabilities previously enabled is not longer enabled.

4. The capability mechanism supports an acknowledgement mechanism for use with capabilities that effect the manner in which LDP messages are constructed by a LDP speaker and processed by its LDP peer. The purpose of the mechanism is to ensure that when a protocol message is received during the state transition of such a capability the capability state used by the receiver to process the message is the same as the state used by the sender to construct the message.

   To achieve the desired coordination the LDP speaker that initiates the state change for such a capability shall include an acknowledgement request with the capability in the Capability message to its peer, and the peer receiving a Capability message for a change that includes an acknowledgement request must acknowledge its receipt.

   For the LDP speaker initiating the capability change the change for generating messages related to the capability takes effect immediately after the Capability message is sent, and the change for processing received messages takes effect immediately after an acknowledgement is received from its peer. For the peer receiving the capability change announcement the change for processing received messages takes effect immediately after the Capability change is received and the change for generating messages takes effect immediately after the acknowledgement is sent.

3. Examples

This section illustrates use of the LDP capability negotiation mechanism with several scenarios. Each scenario considers two LSR's (A and B) with an LDP session:

```
          A -------- B
```

Example 1: Label Request Message and Wildcard FEC

RFC3036 explicitly prohibits use of the Wildcard FEC with the Label Request message. This example illustrates how the capability mechanism could be used to support an enhancement (WFEC) that permits such use.

```
Scenario:   Session initialization.  Suppose that A and B both
            support use of the Wildcard FEC with Label Request
            messages and that both are configured to have the
            capability enabled.
```

The following illustrates capability advertisement during session establishment to enable use of the Wildcard FEC with Label Request messages.

```
    1. A -> B: Init(CapTLV{<WFEC, enable>, ...})
    2. B -> A: Init(CapTLV{<WFEC, enable>, ...})
```

Each includes a Capability TLV in its Initialization message which announces that it will accept and process Label Request messages for the Wildcard FEC. After the session has been established either may send the other Label Request messages for the Wildcard FEC because each knows that the other supports the Wildcard FEC for Label Request messages.

This capability is asymmetric in nature in the sense that one speaker (say A) may have it enabled and the other (B) may not. In such a situation B would be permitted to send A Label Request messages for the Wildcard FEC but A would not be permitted to send B such messages. In practice if only A had the capability enabled it is unlikely that B would make use of it since if it were capable of generating Label Request messages with the Wildcard FEC it would most likely be willing to accept and process them as well.

Example 2: Label Distribution for specific address families

This example assumes an address family (AF) capability which allows an LDP speaker to announce its preparedness to support label distribution for a specific address family. The address family is a

parameter of the AF capability.

The AF capability is an example of a symmetric capability. Label distribution for an address family is of no use unless both peers have it enabled.

Scenario 1: Suppose at session initialization time A is configured for IPv4 and IPv6 label distribution and B is configured for IPv4 label distribution only.

1. A -> B: Init(CapTLV{<AF, ipv4, enable>, <AF, ipv6, enable>, ...})
2. B -> A: Init(CapTLV{<AF, ipv4, enable>, ...})

During session establishment A informs B that its capabilities for label distribution for IPv4 and IPv6 are both enabled, and B informs A that its capability for label distribution for IPv4 is enabled. After the session has been established both A and B may distribute labels for IPv4 FECs to one another. Although B may send A messages for label distribution for IPv6 A may not send B such messages since B has not enabled its capability for IPv6 label distribution.

Scenario 2: Suppose at some time later an operator on B enables label distribution for IPv6. B would respond by sending A a Capability message announcing it.

1. At B an operator enables IPv6 label distribution
2. B->A: Cap(CapTLV{<AF, ipv6, enable, ackreq>, ...})
3. A->B: Cap(CapTLV(<AF, ipv6, enable, ack>, ...)

The message from B to A signals A that the procedures for IPv6 label distribution have been enabled on B and requests an acknowledgement from A. After B sends the message it will process label distribution messages for IPv6 received from A. Since A had previously enabled its label distribution for IPv6 in its Initialization message B may send A label distribution messages for IPv6. Although A had previously enabled IPv6 label distribution, until it receives the Capability message from B it may not send B label distribution messages for IPv6.

The purpose of the acknowledgement is to ensure that there is no ambiguity in the state of the capability during its change. Strictly speaking, in this scenario where the capability had already been enabled at A at session startup the acknowledgement is not required.

Scenario 3: Suppose at some time following the completion of Scenario 2 label distribution for IPv4 is deconfigured on A.

1. At A an operator disables IPv4 label distribution

```
2. A->B: Cap(CapTLV(<AF, ipv4, disable, ackreq>, ...)
3. B->A: Cap(CapTLV{<AF, ipv4, disable, ack>, ...})
```

The message from A to B signals B that the procedures for IPv4 label distribution (with B) has been disabled on A and requests acknowledgement from B. The acknowledgement from B lets A know that B will not longer send label distribution messages for IPv4 to A. Until A receives the acknowledgement it should process any label distribution messages for IPv4 received from B.

4. Encoding

An LDP speaker may announce the state of its capability settings to its peer:

- As part of session establishment by including in the session Initialization message a Capability List TLV that specifies its capability settings. The Keepalive message from the peer that completes session establishment serves as an implicit acknowledgement for any announced capabilities that may require acknowledgement.

- Anytime after the session has been established by sending a Capability message which includes a Capability List TLV.

The format of the Capability List TLV is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|0| Capability List TLV (IANA)|            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Capabilities                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where Capabilities is a list of 1 or more capabilities each of which has the format:

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |C|E|A|  Rsvd   |     Sequence Number                       . . .
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       . . . (cont)   |     Capability Code (IANA)    | Cap Length    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      |                     Capability Data           +-+-+-+-+-+-+-+-+
      |                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

C bit: The Command Bit indicates whether the sender is announcing a change in state for the specified capability or acknowledging receipt of a state change announcement.

1 - Announcement of a change in capability state
0 - Acknowledgement of an announcement

E bit: The Enable Bit indicates whether the sender is enabling the specified capability. The E-bit is ignored for acknowledgements (C-bit = 0).

1 - Capability is enabled
0 - Capability is disabled

A bit: The Acknowledgement Request Bit indicates that the sender wants the receiver to acknowledge the change in state of the specified capability. The A-bit is ignored for acknowledgements (C-bit = 0).

1 - Request acknowledgement for capability announcement
0 - Acknowledgement not required

Rsvd: Reserved

Sequence Number: Uniquely identifies this capability revision. Used for capability changes that require an acknowledgement. Specified by the LDP speaker initiating the change and included in the acknowledgement by its peer.

Capability Code: Identifies the capability. Capability codes are assigned by IANA.

Cap Length: The length (in octets) of any parameters associated with the capability. Must be 0 for a capability that has no parameters.

Capability Data: Data required to specify parameters, if any, associated with the capability.

The LDP Capability message is used by an LDP speaker to announce and to acknowledge changes in the state of one or more of its capabilities. The format of the Capability message is:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |0|   Capability (IANA)         |            Length             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     Message ID                                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     Capability List TLV                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Capability List TLV: The Capability List TLV specifies capability state changes being announced or acknowledged.

5. Compatiblity with RFC3036

The interoperability of LDP speakers that are RFC3036 compliant with LDP speakers that support the LDP capability mechanism described in this document requires the following:

- When an LDP speaker includes a Capability List TLV in an Initialization message the TLV U bit must be 1. This ensures that if the peer does not support the capability mechanism it will ignore the TLV and allow the session to be established.

- When an LDP speaker sends a Capability message to a peer the message U-bit must be 0. This ensures that an RFC3036 peer that does not support the capability mechanism will respond with a Notification message with the Unknown Message Type status code. The Notification message informs the LDP speaker that its peer does not support the capability mechanism. When it receives such a Notification message the LDP speaker should not send further Capability messages to that peer.

- From the point of view of LDP capabilities an RFC3036 compliant peer has label distribution for IPv4 enabled by default. To ensure compatibility with an RFC3036 LDP peer LDP implementations that support capabilities have label distribution for IPv4 enabled until it is explicitly disabled and assume that their peers do as well.

- Existing LDP enhancements that use an ad hoc mechanism for activation (e.g., [RFC3478], [RFC3479]) may continue to do so.

6. Security Considerations

The security considerations described in [RFC3036] that apply to the base LDP specification apply to the capability mechanism described in this document.

7. IANA Considerations

This document specifies a new LDP Message type and a new LDP TLV type both of which require code points assigned by IANA.

In addition this document creates a new name space (the Capability code contained in the Capability List TLV) that is to be managed by IANA. It may be useful to structure that name space into IANA managed, vendor private, and experimental partitions in a mannner similar to the [RFC3036] partitioning of the message and TLV name spaces.

8. Acknowledgements

The author wishes to thank Enke Chen, Eric Rosen, Vanson Lim, Bin Mo and Ina Minei for their comments.

9. References

9.1. Normative References

[RFC3036] Andersson, L., Doolan, P., Feldman, N., Fredette, A. and B. Thomas, "LDP Specification", RFC 3036, January 2001.

9.2. Informative References

[BGP-DYNAMIC] Chen, E., Sanglii, S., "Dynamic Capabilities for BGP-4", draft-ietf-idr-dynamic-cap-08.txt, Work in Progress, June 2006

[IALDP] Decraene, B., Le Roux, JL., Minei, I, "LDP Extensions for Inter-Area LSPs", draft-decraene-mpls-ldp-interarea-01.txt, Work in Progress, October 2005

[MLDP] Minei, I., Wijnamds, I., Editors, "Label Distribution Protocol

Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", draft-minei-wijnands-mpls-ldp-p2mp-00.txt, Work in Progress, September 2005

[NNHOP] Shen, N., Chen, E., Tian, A. "Discovery LDP Next-Nexthop Labels", draft-shen-mpls-ldp-nnhop-label-02.txt, Work in Progress, May 2005

[PWE] Martini L. Editor. "Pseudowire Setup and Maintenance using the Label Distribution Protocol", draft-ietf-pwe3-control-protocol-17.txt, Work in Progress, June 2005

[RFC3478] Leelanivas, M., Rekhter, Y, Aggarwal, R., "Graceful Restart Mechanism for Label Distribution Protocol (LDP)", RFC 3478, February 2003.

[RFC3479] Farrel, A., Editor., "Fault Tolerance for the Label Distribution Protocol (LDP)", RFC 3479, February 2003.

[UPSTREAM] Aggarwal R., Rekhter Y., Rosen E. "MPLS Upstream Label Assignment and Context Specific Label Space", draft-ietf-mpls-ldp-upstream-00.txt, Work in Progress, February 2006.

10. Author Information

Bob Thomas
Cisco Systems, Inc.
1414 Massachusetts Ave.
Boxborough MA 01719

Full Copyright Statement