

MPLS Working Group  
Internet Draft  
Intended status: Standards Track  
Expires: April 6, 2012

Andre Pelletier  
Cisco Systems, Inc.

Kamran Raza  
Cisco Systems, Inc.  
October 7, 2011

## LDP Bindings Refresh

[draft-pelletier-mpls-ldp-bindings-refresh-00.txt](#)

### Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#). This document may not be modified, and derivative works of it may not be created, except to publish it as an RFC and to translate it into languages other than English.

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

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."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on April 6, 2012.

### Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

---

carefully, as they describe your rights and restrictions with respect to this document.

## Abstract

There are situations when there is a need for performing consistency checks for LDP binding state (address/label bindings) exchanged between LDP speakers. For instance, a state refresh may be required to detect and purge stale bindings received by an LDP speaker, which have resulted from an in-service software upgrade. This document specifies mechanics that allow a sender LDP speaker to enclose the initial binding advertisements (or re-advertisements) between explicit START and END of binding markers, thus helping the receiving LDP speaker to detect and purge any extra/stale binding state previously learnt from the sender. In addition to the definition of new LDP Notification message status codes for bindings refresh, this document also extends LDP base specification by introducing the concept of "Wildcard Address" and a new "Wildcard Address Request" message.

## Table of Contents

<a href="#">1. Introduction .....</a>	<a href="#">3</a>
<a href="#">2. Conventions used in this document .....</a>	<a href="#">4</a>
<a href="#">2.1. External Dependencies .....</a>	<a href="#">4</a>
<a href="#">3. Bindings Refresh Procedures .....</a>	<a href="#">4</a>
<a href="#">3.1. LDP Bindings .....</a>	<a href="#">4</a>
<a href="#">3.2. Triggers: Solicited and Unsolicited.....</a>	<a href="#">5</a>
<a href="#">3.3. Operation .....</a>	<a href="#">5</a>
<a href="#">3.3.1. START/END Marker Rules .....</a>	<a href="#">6</a>
<a href="#">3.3.2. Solicited "Wildcard" Requests .....</a>	<a href="#">7</a>
<a href="#">4. Bindings Refresh Signaling .....</a>	<a href="#">8</a>
<a href="#">4.1. "Bindings Refresh" Capability .....</a>	<a href="#">8</a>
<a href="#">4.2. Label Bindings Refresh .....</a>	<a href="#">9</a>
<a href="#">4.2.1. Label START Marker .....</a>	<a href="#">10</a>
<a href="#">4.2.2. Label END Marker .....</a>	<a href="#">10</a>
<a href="#">4.3. Address Bindings Refresh .....</a>	<a href="#">10</a>
<a href="#">4.3.1. "Wildcard Address" in an "Address List" TLV .....</a>	<a href="#">10</a>
<a href="#">4.3.2. "Wildcard Address Request" message .....</a>	<a href="#">11</a>
<a href="#">4.3.3. Address START Marker .....</a>	<a href="#">12</a>
<a href="#">4.3.4. Address END Marker .....</a>	<a href="#">13</a>
<a href="#">5. Operational Examples .....</a>	<a href="#">13</a>
<a href="#">5.1. Basic Use .....</a>	<a href="#">13</a>
<a href="#">5.2. Background Refresh .....</a>	<a href="#">14</a>
<a href="#">6. Security Considerations .....</a>	<a href="#">14</a>

<a href="#">7. IANA Considerations</a>	<a href="#">14</a>
<a href="#">8. References</a>	<a href="#">15</a>
<a href="#">8.1. Normative References</a>	<a href="#">15</a>
<a href="#">8.2. Informative References</a>	<a href="#">15</a>
<a href="#">9. Acknowledgments</a>	<a href="#">15</a>

## [1. Introduction](#)

There are an increasing number of applications that are being multiplexed over a single shared LDP session, each advertising a distinct set of bindings. These applications may be introduced, removed, or encounter a fault during the extended life of the underlying LDP session. In these situations, it would be useful to have an in-service state refresh mechanism to correct discrepancies that may exist between the LDP speakers.

There are scenarios for established LDP sessions where it would be useful for an LDP speaker to know when its peer has begun re-advertisement of all labels and/or addresses, and when this re-advertisement has completed. This delineation would allow the receiver to perform a consistency check and an in-service state reconcile, in a non-service-impacting manner.

For example, if a discrepancy exists between the advertised state of an LDP speaker, versus the same state cached on a peer LSR, currently, the only means of performing a complete reconcile is to either flap the session, or withdrawal and replay of all state. This re-advertisement or flap can adversely affect the network, and trigger second order failures.

The LDP specification [[RFC5036](#)] provides no mechanism for an LDP speaker to notify a peer LSR when it has begun an unsolicited (re-) advertisement of all labels or address bindings to a peer. This document specifies the use of START and END markers to clearly mark the start and end of binding updates. This in turn helps the receiving LSR to perform a hitless, in-service mark-and-sweep state reconcile. The label binding markers, defined in this document, complement and reuse the "End-of-LIB" notification and its procedures defined in [[RFC5919](#)].

The binding refresh procedures specified in this document introduce new LDP capability in accordance with LDP Capability framework [[RFC5561](#)], a new status code, and optional parameters in an LDP Notification message.

---

To provide a complete refresh and consistency check solution for both sender-driven and receiver-driven LDP speakers, this document allows solicited requests of label and/or address binding. To solicit address bindings, this document also extends LDP base specification [[RFC5036](#)] by defining a "Wildcard Address" in an Address List TLV and a new "Wildcard Address Request" LDP message.

## [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 [RFC-2119](#) [[RFC2119](#)].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying [RFC-2119](#) significance.

In this document, the term "binding", when used unqualified, refers to both label and address binding. The term "START marker", when used unqualified, refers to a notification delimiting the start of label or address advertisement. Likewise, the term "END marker" refers to a notification delimiting the completion of address or label advertisement.

### [2.1.](#) External Dependencies

Implementation of this draft is dependent on the following specifications defined outside this document:

- . The use of Typed Wildcard FEC Element [[RFC5918](#)] mandates that firstly LDP "Typed Wildcard FEC" capability be successfully negotiated between LDP peers before label binding markers can be exchanged.
- . This specification also updates the End-of-LIB notification format as originally defined in [RFC5919](#).

## [3.](#) Bindings Refresh Procedures

To facilitate the description in the following sections, let us consider two LDP speakers R1 and R2 with an LDP session between them.

### [3.1.](#) LDP Bindings

The state exchanged on an established LDP session between LDP peers mainly consists of two types of bindings [[RFC5036](#)]:

1. Label bindings: FEC to Label Mappings
2. Address bindings: Local interface IP addresses

Typically, LDP bindings are associated with an address family, and are advertised and withdrawn using their specific mapping and withdraw LDP messages. As per procedures and messages defined in [RFC 5036](#) and [RFC 5918](#), label bindings can also be solicited (requested).

### [3.2](#). Triggers: Solicited and Unsolicited

Loss or errors in tracking of advertised state within the advertising or receiving LDP speaker can result in indefinite retention of stale and potentially damaging state in the receiver LSR. The root cause of these inconsistencies is outside the scope of this document, but may include:

- o In-service software upgrades
- o Protocol process restarts
- o Stateful switchovers
- o Software defects

In such scenarios, a software or end-user trigger may initiate a state refresh between the peers for reconciling for both Label and Address bindings.

Upon some trigger event, speaker R1 may perform an "unsolicited" outbound state reconcile with peer R2 by pushing all bindings of a given type to R2. Similarly, R2 may request a "solicited" state reconcile from R1 by requesting R1 to replay all bindings of a given type.

It is to be noted that while an LSR can use "Label Request" message to solicit Label binding(s), LDP specification [[RFC5036](#)] contains no provision to request address bindings from a LDP peer. This document introduces a new LDP message, "Wildcard Address Request" message, for soliciting addresses from an LDP peer.

### [3.3](#). Operation

LDP speakers that are capable of performing a binding refresh, as specified in this document, first exchange "Binding Refresh" LDP capability [[Section 4.1](#)]. After successful negotiation of this capability, both LDP speakers MUST respond as specified when receiving messages defined in this specification.

An LDP speaker SHOULD provide a user interface to an LSR administrator to trigger an unsolicited binding refresh/re-advertisement towards a peer or set of peers. A similar user interface SHOULD be provided to solicit bindings refresh from a peer or set of peers.

When advertising (or re-advertising) all its bindings, whether for solicited or unsolicited reasons, an LDP speaker performs the following sequence for a given binding type:

- o Transmit a START marker identifying the given binding type;
- o Advertise all bindings for given type;
- o Transmit an END marker identifying the given binding type.

After END marker is received, the receiving LDP speaker MUST purge any previously received bindings that were not re-advertised within the START/END marker boundaries.

Although this specification defines the START/END markers to enclose bindings advertisement, the specification does not restrict or limit other [RFC5036](#) messaging from occurring during this advertisement period. For example, an LDP speaker may be re-advertising all its label bindings for a certain FEC type as a low-priority background reconcile task, but continues to advertise or withdraw addresses or other FEC type bindings during this time.

NOTE: As a conformance test, if the START and END markers were to be replaced by NO-OPs, the remaining sequence of advertisements and withdrawals must continue to conform to [RFC5036](#).

### [3.3.1](#). START/END Marker Rules

#### [3.3.1.1](#). Sender Rules

- o When advertising all bindings of a given type, the associated START/END markers MUST bound the entire set of bindings of given type.
- o There is no ordering restriction between START and/or END markers of a different binding types.



- o If an LDP speaker is currently performing unsolicited advertisement of bindings, and receives a solicited wildcard request from the peer LDP speaker for the same binding type, then it must abort the current refresh, and must restart the re-advertisement from the beginning -- i.e. START marker, followed by all bindings of the requested type, followed by END marker.
- o An LDP speaker may restart a binding refresh by sending another START marker of the same type, and then re-advertising the bindings from the beginning, followed by the associated END marker -- e.g. a user may interrupt a background refresh by manually requesting another refresh of the same type.

#### 3.3.1.2. Receiver Rules

- o Upon receiving a START marker, the receiver MUST flag all associated bindings as STALE.
- o Upon receiving an END marker, the receiver MUST purge any associated bindings that were not refreshed.
- o Any received START marker for a given binding type MUST be considered by the receiver to supersede any previously received START marker of the same type.
- o Any END marker for a given binding type MUST be considered by the receiver to be paired with the most recently received START marker of the same type.

#### 3.3.2. Solicited "Wildcard" Requests

Upon receiving a Wildcard Label Request or Wildcard Address Request from a peer with "Bindings Refresh" capability already negotiated successfully, an LDP speaker:

- o MAY send a START marker prior to responding with the requested bindings. Returning a START marker is OPTIONAL, as the requesting LSR is able to infer that the request itself is equivalent to a START marker; all requested bindings will implicitly follow the request event.
- o MUST signal completion of the response by sending an END marker.  
When sending an END marker in response to a Wildcard Request, the END marker notification message MUST contain the "Message ID" TLV of the associated Wildcard request. This ensures that the receiver is able to correlate the END marker back to the associated wildcard

request, as opposed to some unrelated END sent as part of an unsolicited re-advertisement. This additional "Message ID" TLV is placed under the "Optional Parameters" section of an LDP Notification message. This specification, thus, also updates the End-of-LIB notification format as originally defined in [RFC5919](#).

The requesting LDP speaker MAY assume that any bindings that were not returned between the request and the END marker containing the associated message ID TLV response are stale, and may be purged.

If an LDP speaker has dispatched a solicited Wildcard Request for a given binding type, and subsequently receives an END marker for the same binding type, the receiver MUST ensure the presence of the "Message ID" TLV within the END marker notification in order to perform a purge of stale entries. The receiver LDP speaker MUST notify the sender by transmitting LDP Notification message with "Missing Message Parameters" status code if no such TLV is found. Conversely, if the LDP speaker has solicited a Wildcard request and receives an unsolicited END marker for the same binding type (containing no "Message ID" TLV), this marker must be discarded, and the receiver MUST NOT perform any sweep of stale entries.

#### [4.](#) Bindings Refresh Signaling

##### [4.1.](#) "Bindings Refresh" Capability

"Bindings Refresh" capability is a new LDP capability, defined in accordance with LDP Capability definition guidelines [[RFC5561](#)].

An LDP speaker advertises "Bindings Refresh" capability to announce to its peer its capability of supporting consistency check procedures specified in this document. This capability MAY be sent either in an Initialization message at the session establishment time, or in a Capability message dynamically during the lifetime of a session (only if "Dynamic Announcement" capability [[RFC5561](#)] has been successfully negotiated).

The format of this capability is as follows:



through LDP Notification message with following constructs:

- o A status TLV (with TLV E- and F-bits set to zero) that carries a "Start-of-LIB" Status Code.
- o A FEC TLV with a single Typed Wildcard FEC Element [[RFC5918](#)] that identifies the FEC type for the label bindings those are to follow. In terms of [Section 3.5.1 of RFC5036](#), this TLV is an "Optional Parameter" of the Notification message.

#### [4.2.2](#). Label END Marker

The label "END" marker marks the end of label (re-) advertisement towards a peer. This specification re-uses the "End-of-LIB" notification defined in [RFC5919](#) to implement this marker. Given that this document already mandates the successful negotiation of "Bindings Refresh" capability before using End-of-LIB notification, the document does not further require/mandate additional negotiation of "Unrecognized Notification" Capability with peer [[RFC5919](#)] for End-of-LIB support.

In addition to marking the completion of initial label advertisement procedure defined in [RFC5919](#), this marker MUST also be sent upon completion of an unsolicited re-advertisement, or upon completion of solicited typed wildcard requests, of all label bindings of given FEC type.

#### [4.3](#). Address Bindings Refresh

An LDP speaker that has successfully negotiated the "Bindings Refresh" capability with a peer signals the commencement and end of its address (re-) advertisements to the peer by means of a Notification message as described in following subsections. New constructs are first defined that are required to signal START and END markers for address bindings.

##### [4.3.1](#). "Wildcard Address" in an "Address List" TLV

[RFC5036](#) defines "Address List TLV" (in [Section 3.4.3](#)) as mandatory TLV for use in an "Address" or "Address Withdraw" message, but it does not define any "Wildcard Address" that can be specified in this TLV.

In the context of this specification, we define a "Wildcard Address" that is specified by an empty "Address List" TLV - i.e. the TLV

containing only the Address-family identifier, with no addresses in it. When received in an address message, it must be treated as "All-addresses" for the given Address-family type.

The "Wildcard Address" is to be used in "Wildcard Address Request" message as defined later in the document. This specification, however, does not limit the applicability of "Wildcard Address" and allows it to be used in an Address Withdraw message as well.

#### 4.3.2. "Wildcard Address Request" message

[RFC5036](#) specifies mechanisms and messages to solicit label bindings from peer using Label Request message, but does not define any soliciting mechanisms for address bindings.

In this document, a new LDP message type "Wildcard Address Request" is being defined. The format of Wildcard Address Request message is as follows:

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U| Wcrd Address Request(IANA) |           Message Length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Message ID                               |

```

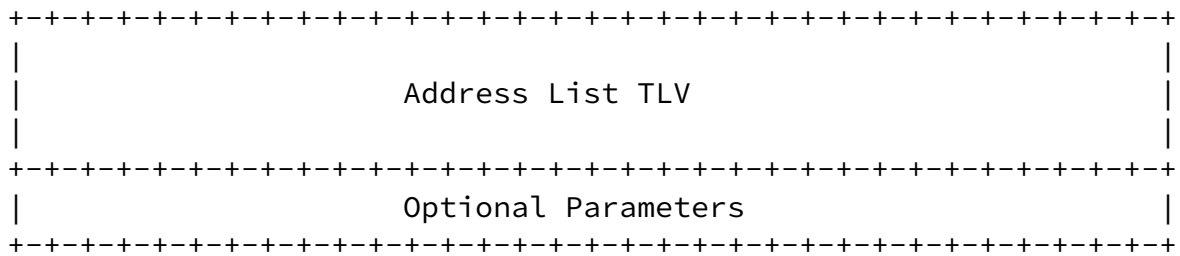


Figure 2 : Wildcard Address Request message format

Where:

U-bit: MUST be 0.

Message ID: 32-bit value used to identify this message [[RFC5036](#)].

Address List TLV: TLV and its encoding as specified in [[RFC5036](#)]. This TLV MUST contain an empty address list (i.e. "Wildcard Address" as defined in this document [section 4.3.1](#)).

Optional Parameters: No optional parameters are defined for the Address message.

Having negotiated "Bindings Refresh" capability, an LDP speaker MUST respond to received "Wildcard Address Request" message by replaying all its address bindings in one or more Address messages to the requesting LSR. When an Address message contains address bindings being sent as a response to incoming Wildcard Address Request message, the Address message MUST contain the "Message ID" TLV containing the message id of the request message.

### [4.3.3](#). Address START Marker

The Address "START" marker marks the commencement of address (re-) advertisement towards a peer for given address family. This marker is signaled to an LDP peer through an LDP Notification message with the following constructs:

- o A status TLV (with TLV E- and F-bits set to zero) that carries a "Start-of-Addresses" Status Code (value to be assigned by IANA).
- o A single "Address List" TLV with given address family and "Wildcard Address". In terms of [Section 3.5.1 of RFC5036](#), this TLV is an "Optional Parameter" of the Notification message.

### [4.3.4](#). Address END Marker

The Address "END" marker marks the end of address (re-) advertisement towards a peer for a given address family. This marker is signaled to an LDP peer through Notification message with following constructs:

- o A status TLV (with TLV E- and F-bits set to zero) that carries a "End-of-Addresses" Status Code (value to be assigned by IANA).
- o A single "Address List" TLV with given address family and "Wildcard Address". In terms of [Section 3.5.1 of RFC5036](#), this TLV is an "Optional Parameter" of the Notification message.

## [5. Operational Examples](#)

### [5.1. Basic Use](#)

A basic use-case would consist of:

```
Advertise X1
Advertise X2
Advertise X3
START (binding type X)
Advertise X1
Advertise X2
END (binding type X)
```

In the above sequence the receiver would create database entries upon initially receiving X1, X2 and X3.

Upon receiving the START marker, the receiver would flag all previously received bindings of type X as stale. The re-advertisement of bindings X1 and X2 would cause the receiver to flag these database entries as refreshed.

Upon receiving the END marker, the receiver would purge binding X3, as it was not refreshed.

### [5.2. Background Refresh](#)

In this example, a background binding refresh is interrupted by a withdrawal, and advertisement of a new binding:

```
Advertise X1
Advertise X2
Advertise X3
START (binding type X)
Advertise X1
```

```
Advertise X2
>> Withdraw X1
Advertise X3
>> Advertise X4
END (binding type X)
```

In the above sequence, all bindings were refreshed, but binding X1 was withdrawn during the re-advertisement sequence, and another binding X4 was introduced.

The final receiver database must contain only bindings X1, X2, X3 and X4. In this example, no bindings were purged upon receiving the END marker.

## [6. Security Considerations](#)

This extension to LDP does not introduce any new security considerations beyond that already apply to the base LDP specification [[RFC5036](#)] and [[RFC5920](#)].

## [7. IANA Considerations](#)

The document introduces following new protocol elements that require code point assignment by IANA:

- o "Bindings Refresh Capability" TLV (requested code point: 0x50F from LDP registry "TLV Type Name Space")
- o "Wildcard Address Request" message (requested code point: 0x302 from LDP registry "Message Type Name Space").
- o New LDP status codes (requested code points as follows, to be allocated from the LDP registry "Status Code Name Space"):

Range/Value	E	Description
-----	---	-----
0x00000031	0	Start-of-LIB
0x00000032	0	Start-of-Addresses
0x00000033	0	End-of-Addresses

## [8. References](#)

### [8.1. Normative References](#)



- [RFC5036] Andersson, L., Doolan, P., Feldman, N., Fredette, A. and Thomas, B., "LDP Specification", [RFC 5036](#), January 2001.
- [[RFC5919](#)] R. Asati, P. Mohapatra, E. Chen, B. Thomas, "Signaling LDP Label Advertisement Completion", [RFC 5919](#), August 2010.
- [[RFC5918](#)] Asati, R., Minei, I., and Thomas, B. "Label Distribution Protocol Typed Wildcard FEC", [RFC 5918](#), August 2010.
- [[RFC2119](#)] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [[RFC5561](#)] Thomas, B., Raza, K., Aggarwal, S., Aggarwal, R., and JL. Le Roux, "LDP Capabilities", [RFC 5561](#), July 2009.

## [8.2](#). Informative References

- [RFC5920] Fang, L. et al., "Security Framework for MPLS and GMPLS Networks", [RFC 5920](#), July 2010.

## [9](#). Acknowledgments

The authors would like to acknowledge Eric Rosen for his review and input on this specification.

This document was prepared using 2-Word-v2.0.template.dot.

### Authors' Addresses

Andre Pelletier  
Cisco Systems, Inc.  
2000 Innovation Drive,  
Ottawa, Ontario K2K-3E8, Canada.  
Email: [apelletti@cisco.com](mailto:apelletti@cisco.com)

Kamran Raza  
Cisco Systems, Inc.  
2000 Innovation Drive,  
Ottawa, Ontario K2K-3E8, Canada.  
Email: [skraza@cisco.com](mailto:skraza@cisco.com)

