

IETF 86 - Orlando  
March 2013

# LDP Bindings Refresh

(draft-pelletier-mpls-ldp-bindings-refresh-02)

André Pelletier

Kamran Raza

(Cisco Systems, Inc.)

# Problem

- \* There are situations when there is a need for performing consistency checks and state refresh 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.
- \* With introduction of high availability features such as NSR, it is possible to preserve the TCP session across in-service-software-upgrades, hardware failovers, or process crashes.
- \* When such an HA event occurs (i.e. without flapping the TCP session), an LSR can re-advertise all local state to the peers (optimizations possible)

# Problem (cont'd)

- \* Full re-advertisement of all state after an HA event does not remove any stale bindings being held by the peer LSR
- \* If an LSR loses track of a piece of advertised/withdrawn state when an HA event occurs, it is possible that a withdraw will never be sent
  - \* The receiver will be stuck holding this state indefinitely
  - \* This stale state can cause future harm, like an address mapping
- \* Tracking every piece of state to standby instance is complex:
  - \* Must be synchronized with respect to TCP stream
  - \* Compounded by various LDP applications (mLDP, AToM)

# Solution: LDP Binding Refresh

- \* RFC5919 introduced “END-of-LIB” marker which can be used to signal completion of a replay
- \* When an HA event occurs and TCP session is preserved, it is possible to re-advertise all bindings, and signal END-of-LIB, but this won't trigger a receiver to clear stale state
- \* This draft proposes a simple mark-and-sweep solution:
  1. START-MARKER
  2. Replay all state...
  3. END-MARKER
    - \* Stale state flushed by the receiver

# Solution: LDP Binding Refresh (cont'd)

- \* The markers allow an LSR to PUSH a state refresh to a peer, thus triggering any stale state clearance.
- \* Any state which is not re-advertised between the markers must be assumed to be stale, and should be purged by the receiver.
- \* State = Label and Address Bindings

	Labels	Addresses
<b>START Marker</b>	Start-of-LIB	Start-of-Addresses
<b>END Marker</b>	End-of-LIB (Existing RFC5919)	End-of-Addresses

- \* For receiver-driven refresh and consistency check, solicited requests of label and/or address binding is also allowed.

# State Refresh Triggers

- \* With the control messages defined in this draft, an LDP LSR can push and pull a full state refresh to correct inconsistencies due to:
  - \* In-service software upgrades (ISSU)
  - \* Protocol process failures and restarts
  - \* Stateful switchovers
  - \* Software defects
- \* In addition, an end-user could also trigger a full state reconcile between LDP LSRs without flapping the TCP sessions.



# Protocol Extensions (cont'd)

- \* Markers:
  - \* Label START:
    - \* LDP Notification message with (a) Status TLV ("Start-of-LIB"), (b) FEC TLV (Typed Wildcard FEC element)
  - \* Label END:
    - \* No change, defined in RFC 5919 [ End-of-LIB ]
  - \* Address START:
    - \* LDP Notification message with (a) Status TLV ("Start-of-Addresses"), (b) AddressList TLV ("Wildcard Address")
  - \* Address END:
    - \* Same as above



## I-D Status

- \* Open Items:

- \* AToM/mLDP/ICCP application state reconcile ?

- \* Next Steps:

- \* Seeking WG feedback
- \* Looking for WG adoption