IETF 81 – Quebec City
July 2011

LDP Outbound Label Filtering

(draft-raza-mpls-ldp-olf-00.txt)

Kamran Raza
Sami Boutros
Pradosh Mohapatra

(Cisco Systems, Inc.)

Problem

- In Downstream Unsolicited label advertisement mode, an LDP speaker LSR1 may receive label bindings from peer LSR2 for the FECs in which it has no interest.
- The label bindings are sent by the peer LSR2, received at the LDP speaker LSR1, and then discarded.
- This is wasteful from resource point of view:
  - Sender LSR resources
  - Network resources
  - Receiver LSR resources

Solution: Receiver-pushed OLF Policy at Sender

- One possible solution is to use DoD mode:
  - Solicit label bindings from peer for interesting FECs only.
  - The drawbacks:
    - DoD mode mandated for both LSRs though it's really required by one of them.
    - Slower convergence compared to DU mode.
- This draft proposes an alternative solution:
  - Receiver LSR1 pushes its label policy to sender LSR2 that LSR2 applies before sending FEC-label bindings towards LSR1.
  - Label policy is pushed/applied for initial updates as well as updated dynamically.

Operational Examples

- LSR with limited LIB space
  - Use OLF framework to initially disable all label binding exchange at the peer side, and then selectively allow FEC via OLF filter updates sent to peer
- Label Filtering at ABRs
  - ABR1 (backbone and non-zero areas) LDP speaker may advertise bindings for prefixes from all area, when ABR2 LDP requires bindings only for backbone area prefixes.
  - Use OLF mechanisms to filter unnecessary prefix label bindings towards ABR2 at ABR1.

OLF Mechanism

- LSRs exchange "OLF Capability" to announce their capability to send or receive label policy filters. The capability can be:
  - announced at session establishment time in LDP Initialization message
  - announced/withdrawn dynamically during session life time if LSRs support "Dynamic Capability Announcement" capability.
- After capability negotiation, the label policy and its update is sent in an LDP Notification message with new "OLF Status".
  - No "incremental" updates – new filter replaces previous filter.
  - Post Capability negotiation:
    - "receive" capable LSR: No FEC-label bindings sent to peer and wait for policy message
    - "send" capable LSR: Push its label policy filter to peer.
  - Label Policy update:
    - "send" capable LSR: Constructs OLF filter according to local policy and push to peer.
    - "receive" capable LSR: Applies received OLF filter and announces/withdraws label bindings accordingly from the peer.

OLF Mechanism (2)

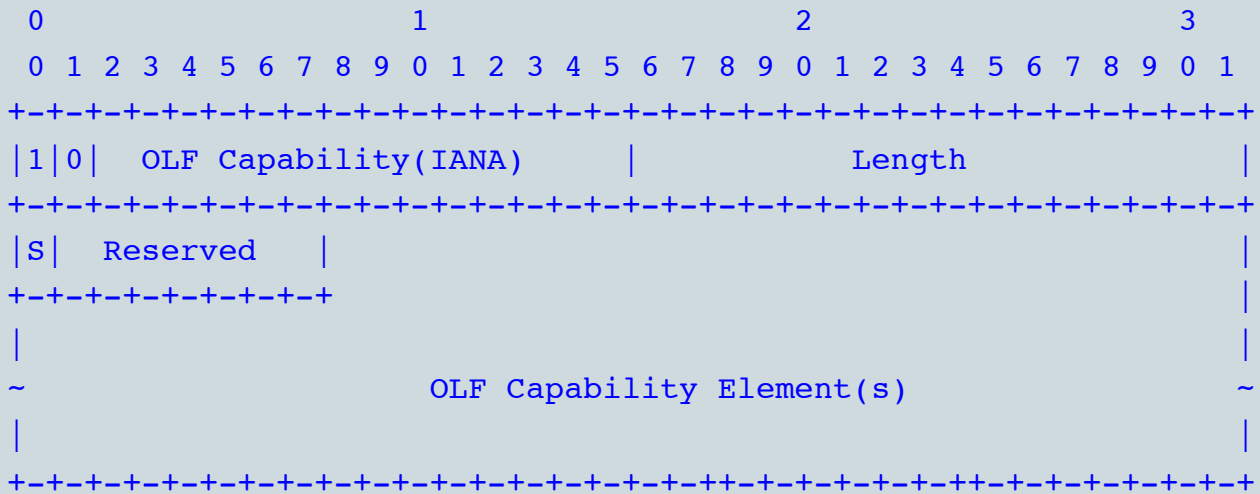
- The OLF capability negotiation, and the policy updates are performed, for a given FEC type.
  - The actual format of a policy filter and its matching rules are FEC-dependent and are to be specified by FEC designers.
  - This document specifies the format and matching rules of policy filter for “Address Prefix” FEC type.

OLF Policy Framework

- Constructs:
  - OLF-Policy: <OLF-Elements>
  - OLF-Element: <FEC-Type> <OLF-Entries>
    - FEC-Type: <FEC Element Type, Address Family>
  - OLF-Entry: <Action, OLF-value>
    - Action: PERMIT, DENY, PERMIT-ALL
    - OLF-value: FEC-specific component and provides the specification of FEC for matching.
- Rules for OLF Element/Entry:
  - If Action=PERMIT-ALL, no OLF-value component specified.
  - For an OLF Element containing more than one OLF entry, the receiving LSR MUST process the OLF entries in the same order as they are specified inside the OLF element.
  - Each OLF Element has an *implicit* DENY-ALL as the last rule
    - The sender and the receiver of OLF policy to keep this in mind in constructing/ processing an OLF filter respectively.

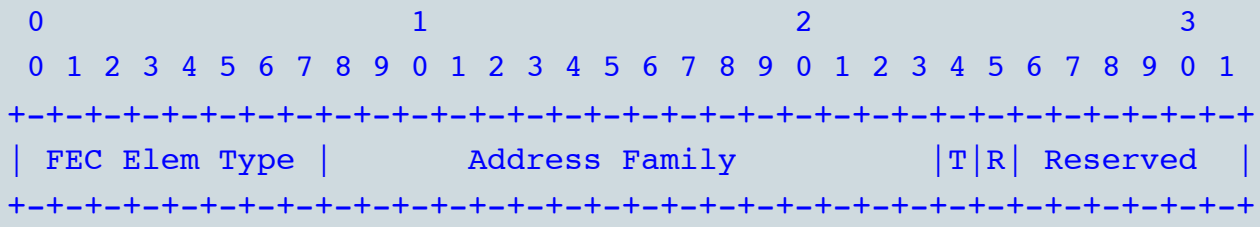
Signaling: OLF Capability

OLF Capability TLV



S=1/0 to announce/withdraw the capability

OLF Capability Element



T: OLF Send capable, R: OLF Receive capable

An LDP speaker that advertises OLF capability MUST support "OLF Policy Status" and "OLF Status" Status Code.

Signaling: OLF Policy

- OLF policy is signaled to a peer through LDP Notification messages, where LDP status code is set to “OLF Status” and OLF specs are carried in new status TLV “OLF Policy Status”
- Sample OLF-Policy:

  [OLF-Policy-Status] <Length>

  [OLF-Element1] <FEC-Elem1><AF1><Length>

  [OLF-Entry1] <Action><OLF-value11>

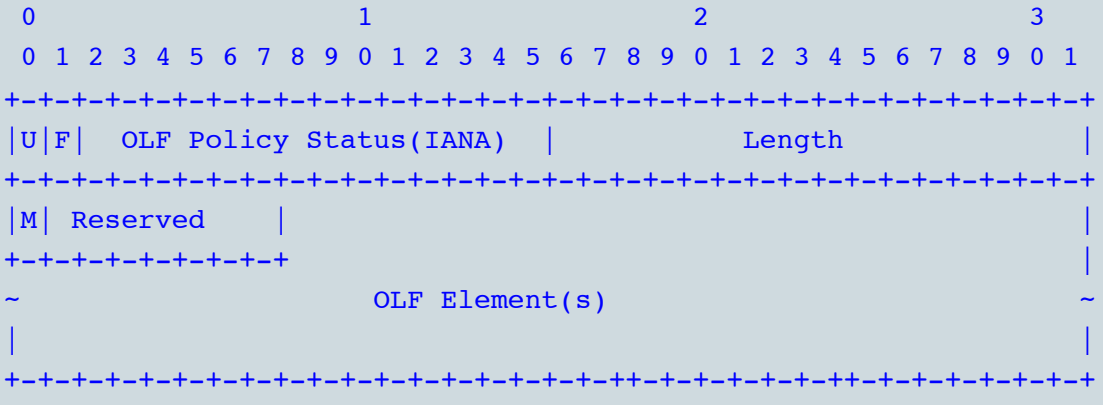
  [OLF-Entry2] <Action><OLF-value12>

  [OLF-Element2] <FEC-Elem2><AF2><Length>

  [OLF-Entry1] <Action><OLF-value21>

  [OLF-Entry2] <Action><OLF-value22>

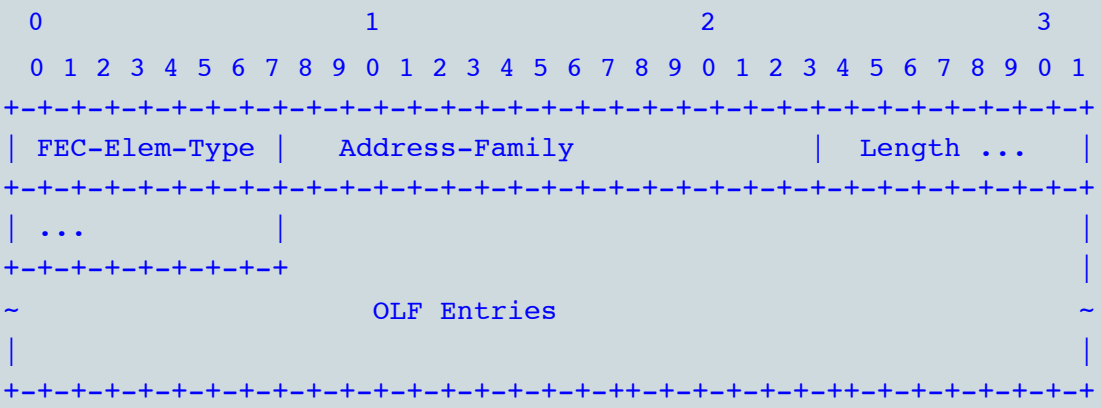
OLF Policy Status TLV



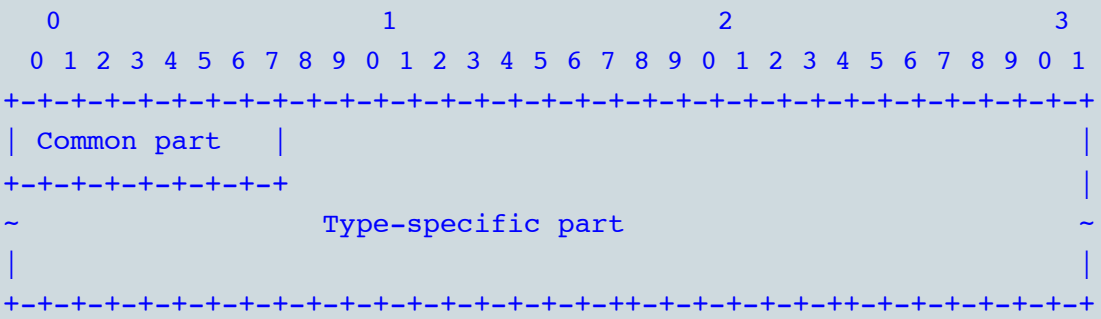
M=“More” bit to indicate if this is partial policy update or end of current update.

Signaling: OLF Policy (2)

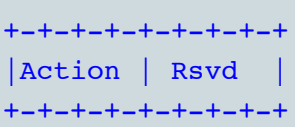
OLF Element



OLF Entry



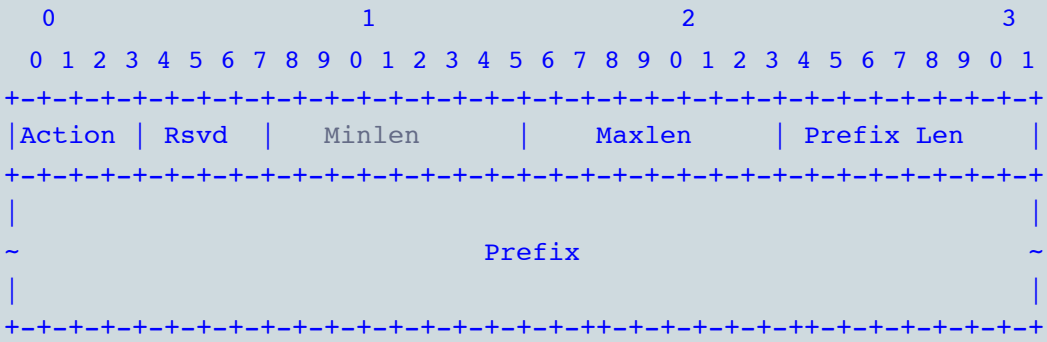
where Common part is



“Address Prefix” FEC OLF Entry

- FEC-Type:
  - FEC-Elem-Type: 0x2 ("Address Prefix")
  - Address-Family: 1 (IPv4) or 2 (IPv6)

Address Prefix FEC OLF Entry



Where:

The "Minlen" and "Maxlen" fields indicate respectively the minimum and the maximum prefix length used for "matching”

“Address Prefix” FEC OLF Entry

- Address Prefix OLF Entry Matching rules:

1. **If** route prefix is neither more specific than, nor equal to, the <Prefix, Prefix Len> fields of the OLF entry: then it is “**NO MATCH**”,
2. **Else**: the route is considered as a **MATCH** to the OLF entry only **IF** following match conditions are met:

OLF Entry Minlen	OLF Entry Maxlen	Route Prefix Match Condition
un-spec.	un-spec.	Route.Prefix Len == OLF.Prefix Len
specified	un-spec.	Route.Prefix Len >= OLF.Minlen
un-spec.	specified	Route.Prefix Len <= OLF.Maxlen
specified	specified	Route.Prefix Len >= OLF.Minlen AND Route.Prefix Len <= OLF.Maxlen

- Sample Filter:

Prefix-IPv4
Permit: [1.1.1.0/24; minlen=31, maxlen=32]
Permit: [2.2.2.2/32; minlen=32, maxlen=32]
Deny: [3.3.3.0/24; minlen=31, maxlen=32]
Permit: [3.3.3.0/24; minlen=24, maxlen=32]
Prefix-IPv6
Permit-All

I-D Status

- Open Items:
  - Change title: LDP Outbound Label Bindings Filtering
  - Provision for OLF filter/policy incremental updates ?
- Next Steps:
  - Seeking feedback
  - Looking for WG adoption
- Acknowledgments:
  - BGP ORF: RFC5291, RFC5292
  - Eric Rosen