IETF 95 – Buenos Aires
April 2016

YANG Data Model for LDP and mLDP

(draft-raza-mpls-ldp-mldp-yang-03)

Kamran Raza	(Cisco)
Rajiv Asati	(Cisco)
Reshad Rahman	(Cisco)
Santosh Esale	(Juniper)
Xufeng Liu	(Ericsson)
Jeff Tantsura	(Ericsson)
Xia Chen	(Huawei)
Loa Andresson	(Huawei)
Himanshu Shah	(Ciena)
Matthew Bocci	(Alcatel-Lucent)

… and several other contributors as acknowledged in the draft

Changes in Rev -03

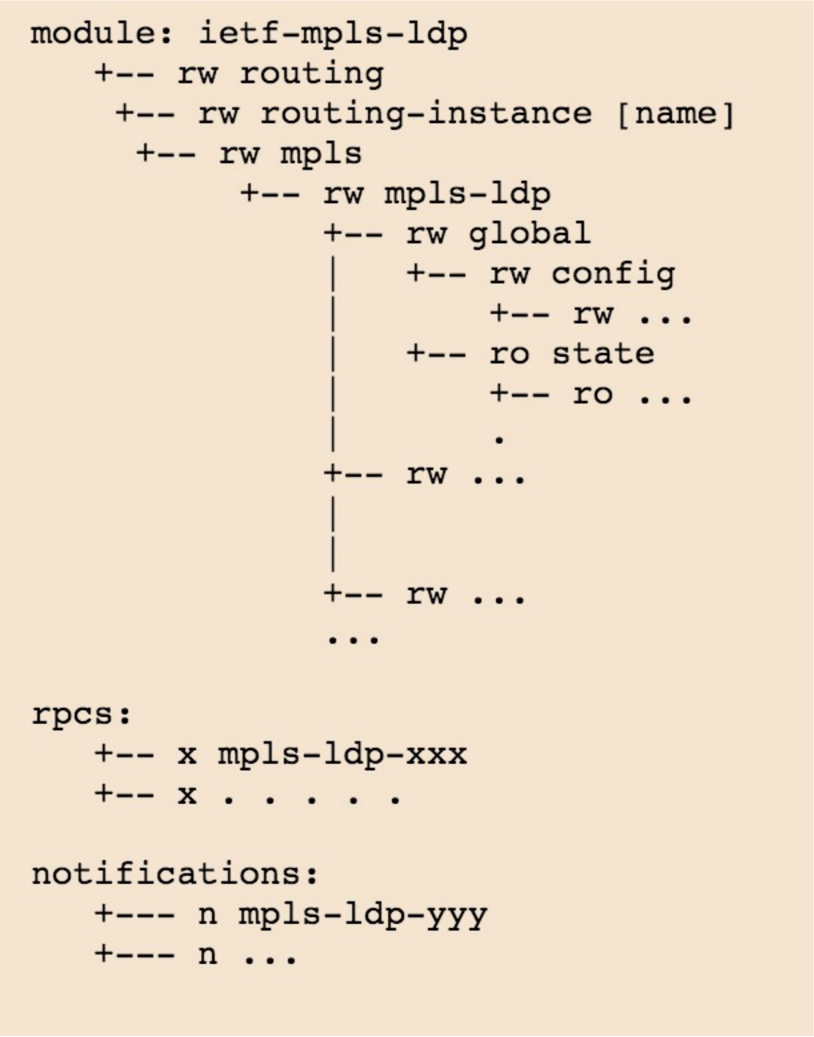
- Addressed **comments from MPLS WG chair** (Ross Callon)
- Added **high-level LDP YANG tree** illustration
- Defined and clarified the use of terms **LDP Neighbor/Adjacency, Session, Peer**
  - Makes changes in the document to adhere to above defs.
- Added **Derived/Operational state** for LDP
- Re-organized and Cleaned up the doc for clarity
- Limited number of authors to 6 on the main page

Ross Callon’s Comments

- Boiler plate rfc2119
- Update “Security Consideration” section
- Clarify IANA section
- Normative references
- Number of authors
- IDNITs

High-level LDP YANG tree

- Better illustration in -03:



```
module: ietf-mpls-ldp
   +-- rw routing
    +-- rw routing-instance [name]
     +-- rw mpls
          +-- rw mpls-ldp
              +-- rw global
              |   +-- rw config
              |       +-- rw ...
              |   +-- ro state
              |       +-- ro ...
              |       .
              +-- rw ...
              |
              |
              +-- rw ...
              ...

rpcs:
   +-- x mpls-ldp-xxx
   +-- x . . . . .

notifications:
   +--- n mpls-ldp-yyy
   +--- n ...
```

Neighbor/Adjacency, Session and Peer

- Our model uses the terms LDP neighbor/adjacency, session, and peer more strictly than RFC 5036 does.

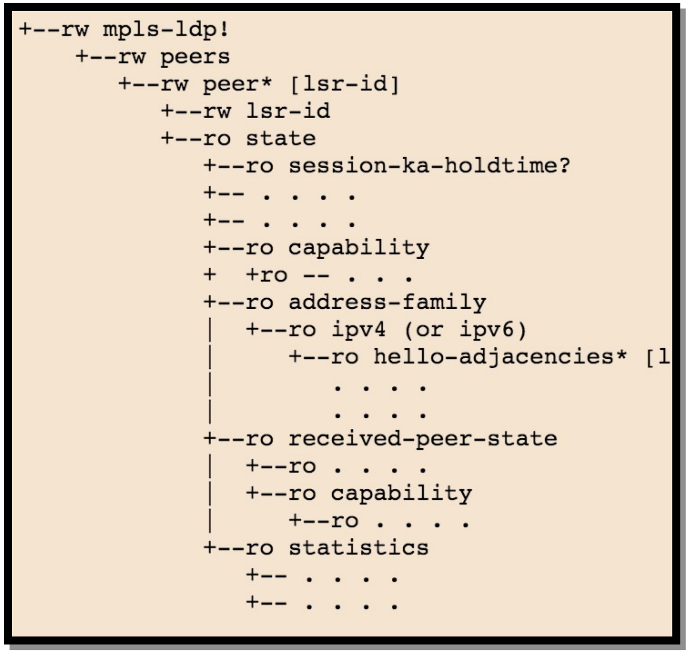
- In this model, these terms are meant as follows:
  - **Neighbor/Adjacency**: An LDP enabled LSR that is discovered through LDP (basic/extended) discovery mechanisms.
  - **Session**: An LDP neighbor with whom a TCP connection has been established
  - **Peer**: An LDP session which has successfully progressed beyond its initialization phase and is ready for binding exchange.

- When used in our model, the above terms refer strictly to the semantics and definitions defined for them.

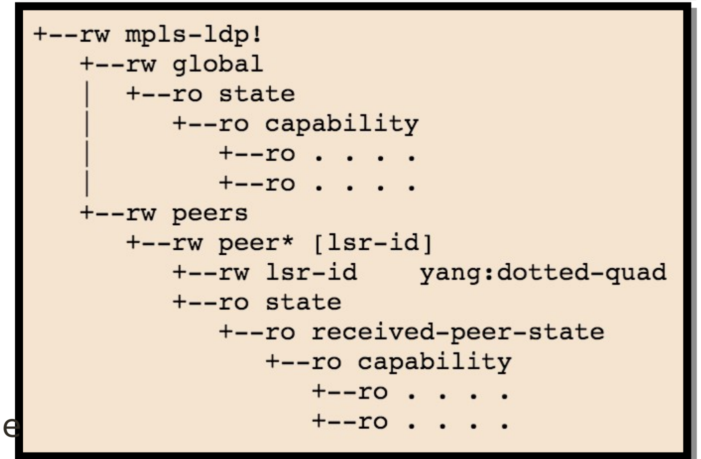
Derived State: LDP

- LDP operational/derived state is defined for 4 areas:

1. Neighbor Adjacencies
2. Peer
3. Bindings:
    1. Address
    2. FEC-label
4. Capabilities



```
+--rw mpls-ldp!
    +--rw peers
       +--rw peer* [lsr-id]
          +--rw lsr-id
          +--ro state
             +--ro session-ka-holdtime?
             +-- . . . .
             +-- . . . .
             +--ro capability
             +  +ro -- . . .
             +--ro address-family
             |  +--ro ipv4 (or ipv6)
             |     +--ro hello-adjacencies* [l
             |        . . . .
             |        . . . .
             +--ro received-peer-state
             |  +--ro . . . .
             |  +--ro capability
             |     +--ro . . . .
             +--ro statistics
                +-- . . . .
                +-- . . . .
```



```
+--rw mpls-ldp!
   +--rw global
   |  +--ro state
   |     +--ro capability
   |        +--ro . . . .
   |        +--ro . . . .
   +--rw peers
      +--rw peer* [lsr-id]
         +--rw lsr-id    yang:dotted-quad
         +--ro state
            +--ro received-peer-state
               +--ro capability
                  +--ro . . . .
                  +--ro . . . .
```

Note that “applied” state is not presented here as it has been covered as part of intended (configuration) state.

Note: See additional slides section for complete “derived state” yang tree

Cross-WG Item Closure

- LSR-Id
  - dotted-quad vs uint32 vs ip-address !!
  - Consistency with other routing protocol router-id

TO-DO List/Items

- Complete Operational/state model for mLDP
- Ensure Alignment with mpls-base
- Ensure Alignment with netmod WG direction
  - Ensure Alignment with routing module changes : network-instance vs routing-instance.

Next Steps

- Request WG adoption
- Close/Keep up with the TODO list (ongoing)
  - Will continue even after the WG adoption

Backup Slide

LDP: Config Hierarchy

```
module: ietf-mpls-ldp
       +-- routing
        +-- routing-instance [name]
         +-- mpls
             +-- mpls-ldp
                +-- global
                |   +-- ...
                |   +-- ...
                |   +-- address-family* [afi]
                |      +-- . . .
                |      +-- . . .
                |   +-- discovery
                |      +-- . . .
                +-- peers
                    +-- ...
                    +-- ...
```

Derived State: LDP Nbr/Adj (1)

Firstly, High level organization of the Nbr/Adj state:

```
+--rw mpls-ldp!
    +--rw discovery
      +--rw interfaces
      |  +--rw interface* [interface]
      |    +--rw address-family* [af]
      |      +--ro state
      |        +--ro ipv4 (or ipv6)
      |          +--ro hello-adjacencies* [adjacent-address]
      |            +--ro adjacent-address
      |              . . . .
      |              . . . .
      +--rw targeted
        +--rw address-family* [afi]
           +--rw afi      address-family
             +--ro state
               +--ro ipv4 (or ipv6)
                 +--ro hello-adjacencies* [local-address adjacent-address]
                   +--ro local-address
                   +--ro adjacent-address
                     . . . .
```

Derived State: LDP Nbr/Adj (2)

Following are nbr/adj state attributes:

```
+--ro hello-adjacencies* [adjacent-address]
          +--ro adjacent-address    inet:ipv4-address
          +--ro flag*               identityref
           +--ro hello-holdtime
                              +--ro adjacent?     Uint16
                    +--ro negotiated?   Uint16
                    +--ro remaining?    Uint16
          +--ro next-hello?        Uint16
          +--ro statistics
                     +--ro discontinuity-time    yang:date-and-time
                     +--ro hello-received?      yang:counter64
                     +--ro hello-dropped?       yang:counter64
           +--ro peer?              Leafref
```

Derived State: LDP Peer (1)

Firstly, High level organization of the peer state:

```
+--rw mpls-ldp!
    +--rw peers
      +--rw peer* [lsr-id]
        +--rw lsr-id
        +--ro state
          +-- . . . .
          +-- . . . .
          +--ro capability
          |    +--. . . .
          +--ro address-family
          |  +--ro ipv4 (or ipv6)
          |    +--ro hello-adjacencies* [local-address adjacent-address]
          |      +-- . . . .
          +--ro received-peer-state
          |  +-- . . . .
          |  +--ro capability
          |    . . . .
          +--ro statistics
            +-- . . . .
            +-- . . . .
```

Derived State: LDP Peer (2)

Following are peer state attributes:

```
+--rw peers
      +--rw peer* [lsr-id]
        +--rw lsr-id    yang:dotted-quad
        +--ro state
          +--ro label-advertisement-mode
          |  +--ro local?       label-adv-mode
          |  +--ro peer?        label-adv-mode
          |  +--ro negotiated?   label-adv-mode
          +--ro next-keep-alive?                  uint16
          +--ro peer-ldp-id?                      yang:dotted-quad
          +--ro received-peer-state
          |  +--ro graceful-restart
          |  |  +--ro enable?          boolean
          |  |  +--ro reconnect-time?   uint16
          |  |  +--ro recovery-time?    uint16
          |  +--ro capability
          |     +--ro end-of-lib
          |     +--ro typed-wildcard-fec
          |     +--ro upstream-label-assignment
```

Derived State: LDP Peer (3)

```
>> CONT’D
        +--ro session-holdtime
        |  +--ro peer?         uint16
        |  +--ro negotiated?   uint16
        |  +--ro remaining?    uint16
        +--ro session-state?                    enumeration
        +--ro tcp-connection
        |  +--ro local-address?    inet:ip-address
        |  +--ro local-port?       uint16
        |  +--ro remote-address?   inet:ip-address
        |  +--ro remote-port?      uint16
        +--ro up-time?                           String
```

Derived State: LDP Peer (4)

```
>> CONT’D
        +--ro statistics
           +--ro discontinuity-time          yang:date-and-time
           +--ro sent (and received)
           |  +--ro total-octets?          yang:counter64
           |  +--ro total-messages?        yang:counter64
           |  +--ro address?               yang:counter64
           |  +--ro address-withdraw?      yang:counter64
           |  +--ro initialization?        yang:counter64
           |  +--ro keepalive?             yang:counter64
           |  +--ro label-abort-request?   yang:counter64
           |  +--ro label-mapping?         yang:counter64
           |  +--ro label-release?         yang:counter64
           |  +--ro label-request?         yang:counter64
           |  +--ro label-withdraw?        yang:counter64
           |  +--ro notification?          yang:counter64
           +--ro total-addresses?            uint32
           +--ro total-labels?               uint32
           +--ro total-fec-label-bindings?   uint32
```

Derived State: LDP Bindings (1)

Firstly, High level organization of the binding state:

```
+--rw mpls-ldp!
    +--rw global
      +--rw address-family* [afi]
        +--rw afi       address-family
        +--ro state
          +--ro ipv4 (or ipv6)
            +--ro bindings
              +--ro address* [address]
              |  +--ro address
              |  +--ro direction?   advertised-received
              |  +--ro peer?        leafref
              +--ro fec-label* [fec]
                +--ro fec     inet:ipv4-prefix
                +--ro peer* [direction peer]
                  +--ro direction           advertised-received
                  +--ro peer                leafref
                  +--ro label?              uint32
                  +--ro used-in-forwarding?   Boolean
```

Derived State: LDP Bindings (2)

Example of address binding derived state:

```
Address bindings:
        Addr 1.1.1.1:
          advertised
        Addr 1.1.1.2:
          advertised
        Addr 2.2.2.2:
          received, peer 192.168.0.2
        Addr 2.2.2.22:
          received, peer 192.168.0.2
        Addr 3.3.3.3:
          received, peer 192.168.0.3
        Addr 3.3.3.33:
          received, peer 192.168.0.3
```

Derived State: LDP Bindings (3)

Example of FEC-label binding derived state:

```
FEC-Label bindings:
      FEC 200.1.1.1/32:
        advertised: local-label 16000
          peer 192.168.0.2:0
          peer 192.168.0.3:0
          peer 192.168.0.4:0
        received:
          peer 192.168.0.2:0, label 16002, used-in-forwarding=Yes
          peer 192.168.0.3:0, label 17002, used-in-forwarding=No

      FEC 200.1.1.2/32:
         . . . .

      FEC 201.1.0.0/16:
         . . . .
```

Derived State: LDP Capabilities

- LDP capabilities state comprise two types of information:
  - Global information (such as timer etc)
  - Per-peer information.

```
+--rw mpls-ldp!
   +--rw global
   |  +--ro state
   |     +--ro capability
   |        +--ro . . . .
   |        +--ro . . . .
   +--rw peers
      +--rw peer* [lsr-id]
         +--rw lsr-id    yang:dotted-quad
         +--ro state
            +--ro received-peer-state
               +--ro capability
                  +--ro . . . .
                  +--ro . . . .
```