Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 16, 2015

X. Chen
Z. Li
Huawei Technologies

August 15, 2014

Yang Model for MPLS LDP
draft-chen-mpls-ldp-yang-cfg-00

Abstract

This document defines a YANG data model that can be used to configure and manage MPLS LDP.

Requirements Language

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].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on February 16, 2015.

Copyright Notice

Table of Contents

1. Introduction

YANG [RFC6020] is a data definition language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF[RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces(e.g. ReST) and encoding other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interface, such as CLI and Programmatic APIs.

This document defines a YANG data model that can be used to configure and manage MPLS LDP. It includes the core LDP[RFC5036] and mLDP[RFC6388]. In addition, features described in different separate MPLS LDP RFC are also supported.

2. Terminology

LDP: Label Distribution Protocol

mLDP: Multipoint extensions for LDP

GR: Graceful Restart

GTSM: Generalized TTL Security Mechanism

3. Design of Data Model

3.1. Overview

The LDP Yang module is divided into two main containers :

o ldpGlobalPara : that contains global writable configuration objects.

o ldpInstances : that contains per-instance writable configuration objects.

The figure below describes the overall structure of the LDP Yang module :

```
module: mplsldp
   +--rw mplsLdp
      +--rw ldpGlobalPara
      |  +--rw grEnable?                    boolean
      |  +--rw reconnectTime?               uint32
      |  +--rw recoveryTime?                uint32
      |  +--rw peerLiveTime?                uint32
      |  +--rw backOffInitTime?             uint32
      |  +--rw backOffMaxTime?              uint32
      +--rw ldpInstances
         +--rw ldpInstance* [vrfName]
            +--rw vrfName             string
            +--rw lsrid               inet:ipv4-address
            +--rw igpSyncDelayTime?   uint32
            +--rw ldpInterfaces
            |  ...
            +--rw ldpRemotePeers
            |  ...
            +--rw ldpPeers
            |  ...
            +--rw ldpGtsms
            |  ...
            +--rw mldpP2mpTunnels
            |  ...
            +--rw mldpP2mpLeafLsps
               ...
```

3.2. LDP Global Configuration

LDP global configuration container includes the global parameters such as graceful restart, backoff timer, ...

3.3. LDP Per-instance Configuration

LDP per-instance configuration container includes parameters of the public LDP instance or the LDP instance binding a specific VRF. LDP per-instance configuration container is divided into:

- o Per-instance parameters.
- o Per-interface configuration of the LDP instance
- o Remote peer configuration of the LDP instance
- o Peer configuration of the LDP instance
- o GTSM[RFC6720] configuration of the LDP instance
- o mLDP P2MP tunnel configuration of the LDP instance
- o mLDP P2MP Leaf LSP configuration of the LDP instance

3.3.1. Per-instance Parameters

The per-instance parameter includes the name of the VRF bound by the LDP instance, LSR ID and timer parameters for LDP IGP Synchronization[RFC5443].

3.3.2. Per-interface Configuration of LDP Instance

Per-interface configuration of the LDP instance includes the interface name, hello timer parameters, keepalive timer parameters, timer parameters for IGP LDP synchronization, transport address.

```
            +--rw ldpInterfaces
            |  +--rw ldpInterface* [ifName]
            |     +--rw ifName                    ifName
            |     +--rw helloSendTime?            uint16
            |     +--rw helloHoldTime?            uint16
            |     +--rw keepaliveSendTime?        uint16
            |     +--rw keepaliveHoldTime?        uint16
            |     +--rw igpSyncDelayTime?         uint32
            |     +--rw transportAddrInterface?   ifName
```

3.3.3. Remote Peer Configuration of LDP Instance

The remote peer configuration of the LDP instance includes the name and the remote IP address of the remote peer, hello timer parameters, keepalive timer parameters, timer parameters for IGP LDP synchronization,

```
+--rw ldpRemotePeers
|  +--rw ldpRemotePeer* [remotePeerName]
|     +--rw remotePeerName             string
|     +--rw remoteIp?                  inet:ipv4-address
|     +--rw helloSendTime?             uint16
|     +--rw helloHoldTime?             uint16
|     +--rw keepaliveSendTime?         uint16
|     +--rw keepaliveHoldTime?         uint16
|     +--rw igpSyncDelayTime?          uint32
```

3.3.4. Peer Configuration of LDP Instance

The peer configuration of the LDP instance includes the peer IP address, authentification parameters for the peer. It may also include the policy to control the distribution of label mapping over the peer which will be defined in the future version.

```
+--rw ldpPeers
|  +--rw ldpPeer* [peerid]
|     +--rw peerid         inet:ipv4-address
|     +--rw md5Password?   string
```

3.3.5. GTSM Configuration of LDP Instance

The GTSM configuration includes the peer transport address and hop limit.

```
+--rw ldpGtsms
   +--rw ldpGtsm* [peerTransportAddr]
      +--rw peerTransportAddr    inet:ipv4-address
      +--rw gtsmHops             uint16
```

3.3.6. mLDP P2MP Tunnel Configuration of LDP Instance

In the ingress LSR, mLDP P2MP tunnel should be defined as the entity used for multicast traffic carried by the mLDP P2MP tunnel. The mLDP P2MP tunnel configuration includes the tunnel name, root IP address and the P2MP LSP ID.

```
               +--rw mldpP2mpTunnels
               |  +--rw mldpP2mpTunnel* [tunnelName]
               |     +--rw tunnelName    string
               |     +--rw rootIp?       inet:ipv4-address
               |     +--rw p2mpLspId?    uint32
```

3.3.7. mLDP P2MP Leaf LSP Configuration of LDP Instance

mLDP P2MP leaf LSP configuration includes the P2MP LSP name, root IP address and P2MP LSP ID.

```
               +--rw mldpP2mpLeafLsps
                  +--rw mldpP2mpLeafLsp* [p2mpLspName]
                     +--rw p2mpLspName    string
                     +--rw rootIp?        inet:ipv4-address
                     +--rw p2mpLspId?     uint32
```

4. LDP Yang Module

```
module mplsldp {
    namespace "urn:huawei:params:xml:ns:yang:mplsldp";
    // replace with IANA namespace when assigned - urn:ietf:params:xml:ns:yang:1
    prefix "mplsldp";
  import ietf-inet-types {
    prefix inet;
  }
    organization
      "Huawei Technologies Co., Ltd.";
    contact
      "jescia.chenxia@huawei.com
       lizhenbin@huawei.com";
    description
      "This YANG module defines the generic configuration
     data for LDP, which is common across all of the vendor
     implementations of the protocol. It is intended that the module
     will be extended by vendors to define vendor-specific
     LDP configuration parameters.";
  revision 2014-08-16 {
    description
      "Initial revision.";
  }

  typedef ifName {
        description "ifName is like ethernet1/1/1/1";
        type string {
            length "1..63";
        }
    }
```

```
    container mplsLdp {

        container ldpGlobalPara {

            leaf grEnable {
                description
                  "Specifies the LDP GR capability.
                   Enabling or disabling of GR will cause the re-establishment o
f all LDP sessions.";
                config "true";
                type boolean;
                default "false";
            }
            leaf reconnectTime {
                description
                  "Specifies the value of the LDP session reconnection timer.
                   The value is in seconds.";
                config "true";
                default "300";
                type uint32 {
                    range "3..3600";
                }
            }
            leaf recoveryTime {
                description
                  "Specifies the value of the LSP recovery timer (s).
                   The value is in seconds.";
                config "true";
                default "300";
                type uint32 {
                    range "3..3600";
                }
            }
            leaf peerLiveTime {
                description
                  "Specifies the value of the neighbor keeplive timer (s).
                   The value is in seconds.";
                config "true";
                default "600";
                type uint32 {
                    range "3..3600";
                }
            }
            leaf backOffInitTime {
                description
                  "Specifies the init value of the exponential backoff timer (s)
.
                   The value is in seconds.";
                config "true";
                default "15";
```

```
                type uint32 {
                    range "5..2147483";
                }
            }
            leaf backOffMaxTime {
                description
                  "Specifies the maximum value of the exponential backoff Timer(
s).
                   The value is in seconds.";
                config "true";
                default "120";
                type uint32 {
                    range "5..2147483";
                }
            }
        }

        container ldpInstances {

            list ldpInstance {

                key "vrfName";
                max-elements "unbounded";
                min-elements "0";
                description "Specifies a list of LDP instances.";

                leaf vrfName {
                    description
                      "Name of an LDP instance.
                      If the name string is empty the instance means a public in
stance whose name is _public_.";
                    config "true";
                    //default "_public_";
                    type string {
                        length "0..32";
                    }
                }
                leaf lsrid {
                    description "LSR ID of an instance.";
                    config "true";
                    mandatory "true";
                    type inet:ipv4-address;
                }
                leaf igpSyncDelayTime {
                    description
                      "Specifies the interval at which an interface waits to est
ablish an LSP after an LDP session is set up.
                       The value is in seconds.";
                    config "true";
                    default "10";
                    type uint32 {
```

```
                    range "0..65535";
                }
            }
            container ldpInterfaces {

                list ldpInterface {

                    key "ifName";
                    max-elements "unbounded";
                    min-elements "0";
                    description "Specifies an LDP interface.";

                    leaf ifName {
                        description "Interface name.";
                        config "true";
                        type ifName;
                    }
                    leaf helloSendTime {
                        description
                          "Specifies the value of the Hello packet sending t
imer.
                           The value is in seconds.";
                        config "true";
                        type uint16 {
                            range "1..65535";
                        }
                    }
                    leaf helloHoldTime {
                        description
                          "Specifies the interval value of the Hello hold ti
mer.
                           The value is in seconds.";
                        config "true";
                        type uint16 {
                            range "3..65535";
                        }
                    }
                    leaf keepaliveSendTime {
                        description
                          "Specifies the value of the Keepalive packet sendi
ng timer (s).
                           The value is in seconds.";
                        config "true";
                        type uint16 {
                            range "1..65535";
                        }
                    }
                    leaf keepaliveHoldTime {
                        description
                          "Specifies the value of the Keepalive packet holdi
ng timer (s).
                           The value is in seconds.";
```

```
                            config "true";
                            type uint16 {
                                range "30..65535";
                            }
                        }
                        leaf igpSyncDelayTime {
                            description
                              "Specifies an interval at which an interface waits
 to establish an LSP after an LDP session is set up.
                               The value is in seconds.";
                            config "true";
                            type uint32 {
                                range "0..65535";
                            }
                        }
                        leaf transportAddrInterface {
                            description "Configures an interface address to be u
sed as the transport address.";
                            config "true";
                            type ifName;
                        }

                    }

                }

                container ldpRemotePeers {

                    list ldpRemotePeer {

                        key "remotePeerName";
                        max-elements "unbounded";
                        min-elements "0";
                        description "Specifies a remote LDP neighbor.";

                        leaf remotePeerName {
                            description "Specifies the name of a remote neighbor
.";
                            config "true";
                            type string {
                                length "1..32";
                            }
                        }
                        leaf remoteIp {
                            description "Specifies the IPv4 address of a remote
neighbor.";
                            config "true";
                            type inet:ipv4-address;
                        }
                        leaf helloSendTime {
                            description
                              "Specifies the value of the Hello packet sending t
imer.
```

```
                                The value is in seconds.";
                            config "true";
                            type uint16 {
                                range "1..65535";
                            }
                        }
                        leaf helloHoldTime {
                            description
                              "Specifies the value of the Hello packet holding t
imer (s).
                               The value is in seconds.";
                            config "true";
                            type uint16 {
                                range "3..65535";
                            }
                        }
                        leaf keepaliveSendTime {
                            description
                              "Specifies the value of the Keepalive packet sendi
ng timer.
                               The value is in seconds.";
                            config "true";
                            type uint16 {
                                range "1..65535";
                            }
                        }
                        leaf keepaliveHoldTime {
                            description
                              "Specifies the value of the Keepalive holding time
r.
                               The value is in seconds.";
                            config "true";
                            type uint16 {
                                range "30..65535";
                            }
                        }
                        leaf igpSyncDelayTime {
                            description
                              "Specifies an interval at which an interface waits
 to establish an LSP after an LDP session is set up.
                               The value is in seconds.";
                            config "true";
                            type uint32 {
                                range "0..65535";
                            }
                        }

                    }

                }

                container ldpPeers {
```

```
                    list ldpPeer {

                        key "peerid";
                        max-elements "unbounded";
                        min-elements "0";
                        description "Specifies an LDP peer.";

                        leaf peerid {
                            description "Specifies an LDP peer ID.";
                            config "true";
                            type inet:ipv4-address;
                        }
                        leaf md5Password {
                            description "Specifies an MD5 password.";
                            config "true";
                            type string {
                      length "1..255";
                  }
                        }

                    }

                }

                container ldpGtsms {

                    list ldpGtsm {

                        key "peerTransportAddr";
                        max-elements "unbounded";
                        min-elements "0";
                        description "Specifies a GTSM security attribute.";

                        leaf peerTransportAddr {
                            description "Specifies a GTSM transport address.";
                            config "true";
                            type inet:ipv4-address;
                        }
                        leaf gtsmHops {
                            description "Specifies the maximum number of GTSM ho
ps. The value is an integer ranging from 1 to 255.";
                            config "true";
                            mandatory "true";
                            type uint16 {
                                range "1..255";
                            }
                        }
                    }
```

```
                }

                container mldpP2mpTunnels {

                    list mldpP2mpTunnel {

                        key "tunnelName";
                        max-elements "unbounded";
                        min-elements "0";
                        description "Specifies the mldp p2mp lsp configuration i
n the root node.";

                        leaf tunnelName {
                            description "Mldp p2mp tunnel name in root node whic
h can be used by service.";
                            config "true";
                            type string {
                                length "1..31";
                            }
                        }
                        leaf rootIp {
                            description "Specifies the root ip address of mldp p
2mp lsp.";
                            config "true";
                            type inet:ipv4-address;
                        }
                        leaf p2mpLspId {
                            description "Specifies the LSP ID of mldp p2mp lsp i
f the generic LSP identifier is a type of opaque value element.";
                            config "true";
                            type uint32 {
                                range "1..8192";
                            }
                        }
                    }

                }

                container mldpP2mpLeafLsps {

                    list mldpP2mpLeafLsp {

                        key "p2mpLspName";
                        max-elements "unbounded";
                        min-elements "0";
                        description "Specifies the mldp p2mp lsp configuration i
n the leaf node.";

                        leaf p2mpLspName {
                            description "The name of mldp p2mp lsp configuration
 in the leaf node.";
                            config "true";
                            type string {
                                length "1..31";
```

```
                            }
                        }
                        leaf rootIp {
                            description "Specifies the root ip address of mldp p
2mp lsp.";
                            config "true";
                            type inet:ipv4-address;
                        }
                        leaf p2mpLspId {
                            description "Specifies the LSP ID of mldp p2mp lsp i
f the generic LSP identifier is a type of opaque value element.";
                            config "true";
                            type uint32 {
                                range "1..8192";
                            }
                        }
                    }

                }

            }

        }

    }

}
```

5. IANA Considerations

This document makes no request of IANA.

6. Security Considerations

The data model defined does not create any security implications. This draft does not change any underlying security issues inherent in [I-D.ietf-netmod-routing-cfg].

7. Acknowledgements

8. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 2007.

[RFC5443] Jork, M., Atlas, A., and L. Fang, "LDP IGP Synchronization", RFC 5443, March 2009.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC6388] Wijnands, IJ., Minei, I., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, November 2011.

[RFC6720] Pignataro, C. and R. Asati, "The Generalized TTL Security Mechanism (GTSM) for the Label Distribution Protocol (LDP)", RFC 6720, August 2012.

Authors' Addresses

Xia Chen
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: jescia.chenxia@huawei.com

Zhenbin Li
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: lizhenbin@huawei.com