Multicast Yang Model

draft-zhang-mboned-multicast-yang-model-00

Mboned WG
IETF 101#London

Sandy. Zhang
Linda. Wang
Ying. Cheng

Update:

draft-zhang-mboned-multicast-info-model-02

↓

draft-zhang-mboned-multicast-yang-model-00

According to YANG doctor’s suggestion:

- Draft name changes.
- The YANG model writing improves.

The main content of model is unchanged.
More descriptions are added.
Add some notifications.

Why introduce Multicast Model?

✓Existed multicast YANG models:

PIM IGMP BIER ……

- These models describe different technologies for multicast;
- These models are distributed as separate file and focus on the protocol itself;
- They are device models;
- They cannot describe a high-level multicast information.

➢ Stand at a high level to take advantage of these models to control the multicast network to implement multicast service.

What is Multicast Model?

- Provide a human readability of the whole multicast network;
- Frame different components and correlate them;
- Based on the human readable UML like Class Diagram, instantiate these classes through YANG model;
- Take full advantage of and depend on existed multicast YANG models;
- Open for future multicast technologies;

- http://www.opendaylight.org/

- OpenDaylight is a highly available, modular, extensible, scalable and multi-protocol controller infrastructure built for SDN deployments on modern heterogeneous multi-vendor networks. OpenDaylight provides a model-driven service abstraction platform that allows users to write apps that easily work across a wide variety of hardware and south-bound protocols.

- ODL employs a model-driven approach to describe the network, the functions to be performed on it and the resulting state or status achieved.

- By sharing YANG data structures in a common data store and messaging infrastructure, OpenDaylight allows for fine-grained services to be created then combined together to solve more complex problems. In the ODL Model Driven Service Abstraction Layer (MD-SAL), any app or function can be bundled into a service that is then then loaded into the controller. Services can be configured and chained together in any number of ways to match fluctuating needs within the network.

BIER project in ODL

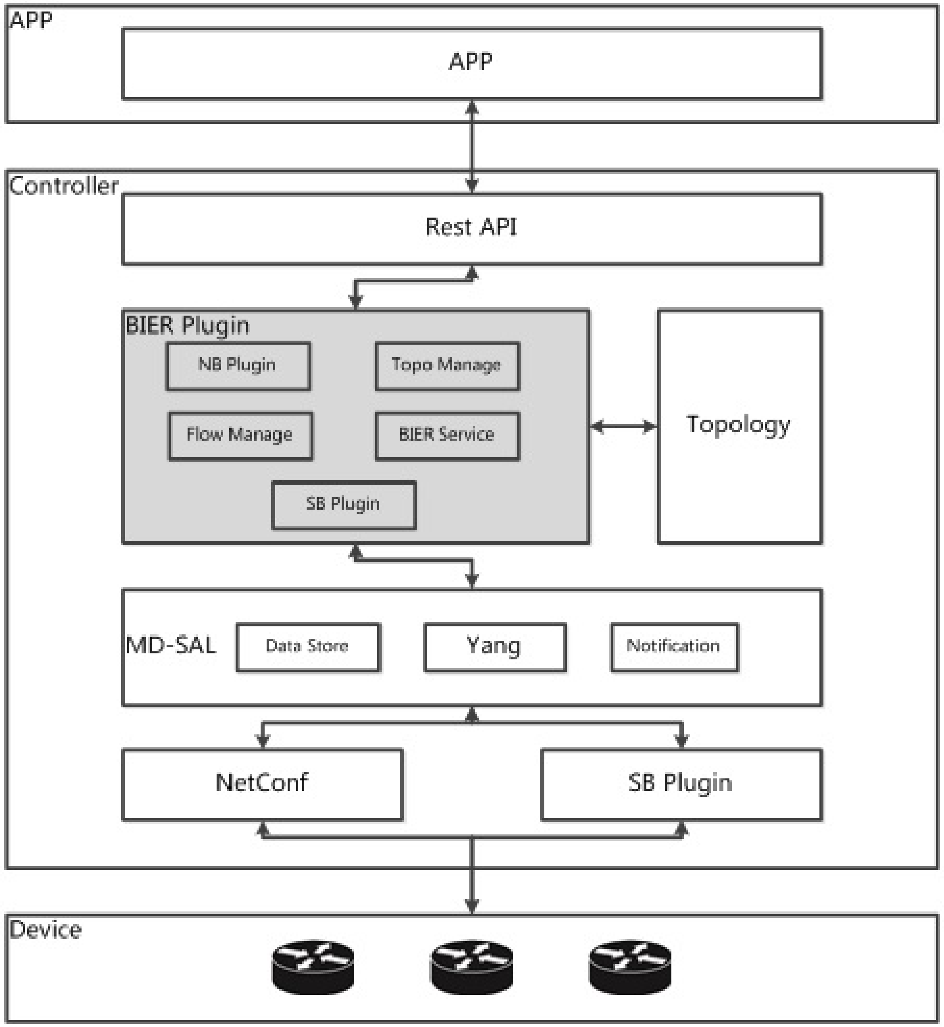
https://wiki.opendaylight.org/view/BIER:Main

The BIER project is driven by two YANG models:

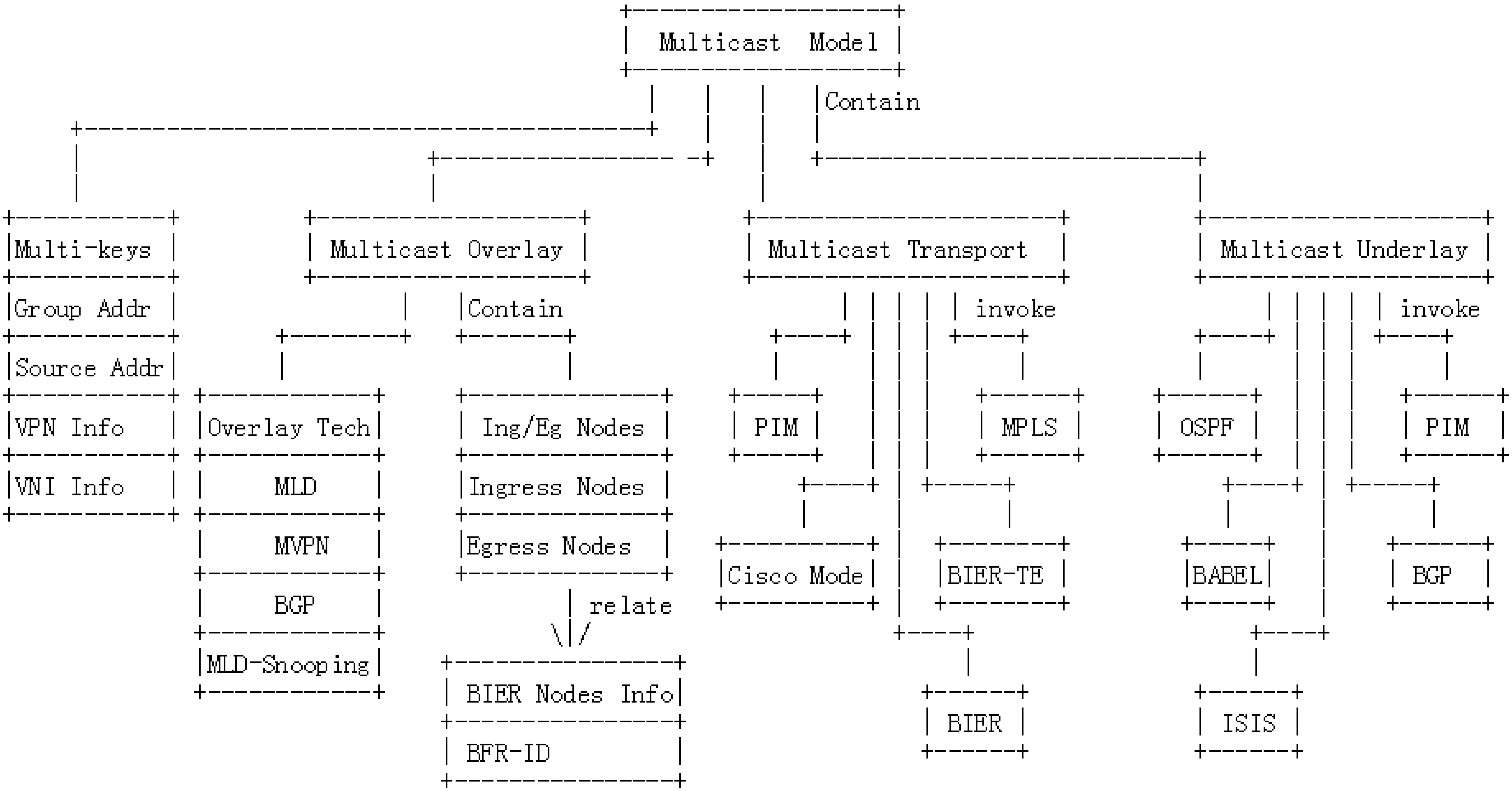
Multicast Model
draft-zhang-mboned-multicast-yang-model-00

YANG Data Model for BIER Protocol
draft-ietf-bier-bier-yang-03

- This model has been verified in ODL BIER project.
- The project had been released in Carbon version.
- This model is feasible and practicable.



Multicast UML like Class Diagram



```
                                            +-------------------+
                                            |  Multicast  Model |
                                            +-------------------+
                                              |   |   |   |Contain
     +----------------------------------------+   |   |   |
     |                         +---------------- -+   |   +--------------------------+
     |                         |                      |                              |
+-----------+         +-------------------+          +----------------------+        +--------------------+
|Multi-keys |         | Multicast Overlay |          | Multicast Transport  |        | Multicast Underlay |
+-----------+         +-------------------+          +----------------------+        +--------------------+
|Group Addr |                |   |Contain                   | | | | | invoke              | | | | | invoke
+-----------+       +--------+   +-------+             +----+ | | | +----+           +----+ | | | +----+
|Source Addr|       |                    |             |      | | |      |           |      | | |      |
+-----------+ +------------+     +--------------+   +-----+   | | |   +------+    +------+  | | |   +------+
|VPN Info   | |Overlay Tech|     | Ing/Eg Nodes |   | PIM |   | | |   | MPLS |    | OSPF |  | | |   | PIM  |
+-----------+ +------------+     +--------------+   +-----+   | | |   +------+    +------+  | | |   +------+
|VNI Info   | |     MLD    |     |Ingress Nodes |        +----+ | +-----+              +----+ | +-----+
+-----------+ +------------+     +--------------+        |      |       |              |      |       |
              |     MVPN   |     |Egress Nodes  |  +----------+ |  +--------+       +-----+   |    +------+
              +------------+     +--------------+  |Cisco Mode| |  |BIER-TE |       |BABEL|   |    | BGP  |
              |     BGP    |             | relate  +----------+ |  +--------+       +-----+   |    +------+
              +------------+            \|/                     +----+                   +----+
              |MLD-Snooping|   +----------------+                    |                   |
              +------------+   | BIER Nodes Info|                 +------+           +------+
                               +----------------+                 | BIER |           | ISIS |
                               | BFR-ID         |                 +------+           +------+
                               +----------------+
```

Multicast Yang Model

```
module: ietf-multicast-model
    +--rw multicast-model
      +--rw multicast-key
          ......
      +--rw multicast-overlay
          ......
      +--rw multicast-transport
          ......
      +--rw multicast-underlay
          ......
```

Overview: Divide the multicast data model into three layers.

Multicast Yang Model

```
+--rw multicast-keys* [vpn-id source-address source-wildcard group-address group-wildcard vni-type vni-value]
   +--rw vpn-id              uint32
   +--rw source-address         inet:ip-address
   +--rw source-wildcard        uint8
   +--rw group-address          inet:ip-address
   +--rw group-wildcard         uint8
   +--rw vni-type            virtual-type
   +--rw vni-value           uint32
```

Multicast keys:

✓ Basic multicast flow information;

✓ Keys of the multicast service.

Multicast Yang Model

```
+--rw multicast-overlay
    |  +--rw nodes-information
    |  |  +--rw ingress-node?   inet:ip-address
    |  |  +--rw egress-nodes* [egress-node]
    |  |     +--rw egress-node    inet:ip-address
    |  +--rw bier-information
    |  |  +--rw sub-domain?     sub-domain-id
    |  |  +--rw ingress-node?   bfr-id
    |  |  +--rw egress-nodes* [egress-node]
    |  |     +--rw egress-node    bfr-id
    |  +--rw overlay-technology
    |     +--rw (overlay-tech-type)?
    |        +--:(mld)
    |        +--:(mvpn)
    |        +--:(bgp)
    |        +--:(mld-snooping)
```

Overlay layer includes:

✓ Ingress/egress nodes information;

✓ Overlay technology.

Multicast Yang Model

```
+--rw multicast-transport
     |  +--rw bier
     |  |  +--rw sub-domain?       sub-domain-id
     |  |  +--rw (encap-type)?
     |  |  |  +--:(mpls)
     |  |  |  +--:(non-mpls)
     |  |  |  +--:(ipv6)
     |  |  +--rw bitstringlength?   uint16
     |  |  +--rw set-identifier?    si
     |  |  +--rw ecmp?              boolean
     |  |  +--rw frr?               boolean
     |  +--rw bier-te
     |  |  +--rw sub-domain?       sub-domain-id
     |  |  +--rw (encap-type)?
     |  |  |  +--:(mpls)
     |  |  |  +--:(non-mpls)
     |  |  +--rw bitstringlength?   uint16
     |  |  +--rw set-identifier?    si
     |  |  +--rw ecmp?              boolean
     |  |  +--rw frr?               boolean
     |  +--rw cisco-mode
     |  |  +--rw p-group?           inet:ip-address
     |  |  +--rw graceful-restart?   boolean
     |  |  +--rw bfd?               boolean
```

Transport layer includes:

- ✓ Transport technology type
- ✓ Corresponding individual YANG models

```
|  +--rw mpls
     |  |  +--rw (mpls-tunnel-type)?
     |  |     +--:(mldp)
     |  |     |  +--rw mldp-tunnel-id?       uint32
     |  |     |  +--rw mldp-frr?             boolean
     |  |     |  +--rw mldp-backup-tunnel?   boolean
     |  |     +--:(p2mp-te)
     |  |        +--rw te-tunnel-id?         uint32
     |  |        +--rw te-frr?               boolean
     |  |        +--rw te-backup-tunnel?     boolean
     |  +--rw pim
     |     +--rw graceful-restart?   boolean
     |     +--rw bfd?               boolean
```

Multicast Yang Model

```
+--rw multicast-underlay
      +--rw underlay-requirement?   boolean
      +--rw bgp
      +--rw ospf
      |  +--rw topology-id?   uint16
      +--rw isis
      |  +--rw topology-id?   uint16
      +--rw babel
      +--rw pim
```

Underlay layer includes:

- ✓ Underlay technology type
- ✓ Corresponding individual YANG models

Multicast Yang Notification

```
notifications:
   +---n head-end-event
     +--ro event-type?          enumeration
     +--ro multicast-key
     |  +--ro vpn-rd?           rt-types:route-distinguisher
     |  +--ro source-address?   ip-multicast-source-address
     |  +--ro group-address?    rt-types:ip-multicast-group-address
     |  +--ro vni-type?         virtual-type
     |  +--ro vni-value?        uint32
     +--ro overlay-tech-type?   enumeration
     +--ro transport-tech?      enumeration
     +--ro underlay-tech?       enumeration
```

Head end event includes:

- Node down/restart
- Module loaded/unloaded

Next Steps

- Any comments ㅆ
- WG adoption?