YANG Data Model for DHCP v4

draft-liu-dhc-dhcp-yang-model-06

Github:
https://github.com/iamChinChen/gallery/blob/master/yang_modeling/ietf-dhcp%402017-03-02.yang

Presenter – Gang Yan, Huawei

Leo Liu Huawei

kunkun Lou Huawei

Chin Chen Ericsson

Introduction

- Defines a YANG data model for DHCP Server, relay, and client, including configuration, operational data, and RPC.

Model Structure

```
 module: ietf-dhcp
      +--rw dhcp
      |  +--rw server {dhcp-server}?
      |  |    ...
      |  |    ...
      |  +--rw relay {dhcp-relay}?
      |  |    ...
      |  |    ...
      |  +--rw client {dhcp-client}?
      |  |    ...
      |  |    ...
      |  +--ro dhcp-state
      |     +--ro server {dhcp-server}?
      |     |  ...
      |     |  ...
      |     +--ro relay {dhcp-relay}?
      |     |  ...
      |     |  ...
      |     +--ro client {dhcp-client}?
      |        ...
      |        ...
rpcs:
   +---x clean-server-statistics     {dhcp-server}?
   +---x clean-relay-statistics      {dhcp-relay}?
   +---x clean-client-statistics     {dhcp-client}?
```

Feature

- Design three if-feature for deployment of DHCP server/relay/client as below:

```
feature dhcp-server {
  description "Feature DHCP server";
}
feature dhcp-client {
  description "Feature DHCP client";
}
feature dhcp-relay {
  description "Feature DHCP relay";
}
```

DHCP Server

Server configurations contain lease time, ping packet, IP address pool and option configuration.
The most important part is the IP pool configuration. Specifying IP address section is for dynamic allocation. Configuring the mapping between IP address and MAC address is for manual allocation.

```
module: ietf-dhcp
 +--rw dchp
         +--rw server
            +--rw lease-time?             uint32
            +--rw ping-packet-number?     uint8
            +--rw ping-packet-timeout?    uint16
            +--rw option
            |  +--…
            +--rw dhcp-ip-pool* [ip-pool-name]
               +--rw ip-pool-name          string
               +--rw interface?            if:interface-ref
               +--rw gateway-ip?           inet:ip-address
               +--rw gateway-mask?         inet:ip-prefix
               +--rw lease-time?           uint32
               +--rw manual-allocation* [mac-address ip-address]
               |  +--rw mac-address     yang:mac-address
               |  +--rw ip-address      inet:ip-address
               +--rw section* [section-index]
               |  +--rw section-index        uint16
               |  +--rw section-start-ip     inet:ipv4-address
               |  +--rw section-end-ip?      inet:ipv4-address
               +--rw option
```

Pre-Define Option

```
module: ietf-dhcp
  +--rw option
     +--rw dhcp-server-identifier?   inet:ip-address
     +--rw domain-name?              string
     +--rw domain-name-server?       inet:ip-address
     +--rw interface-mtu?            uint32
     +--rw netbios-name-server?      inet:ip-address
     +--rw netbios-node-type?        uint32
     +--rw netbios-scope?            string
```

DHCP Relay

The relay function is configured per interface. Enable/disable relay functionality on a specific interface, and specify the DHCP server.

```
module: ietf-dhcp
   +--rw dchp
      +--rw relay
         +--rw server-group* [server-group-name]
            +--rw server-group-name    string
            +--rw interface?           if:interface-ref
            +--rw gateway-address?     inet:ipv4-address
            +--rw server-address*      inet:ipv4-address
```

DHCP Client

DHCP client is also managed per interface, including enable/disable client DHCP client function, client id and lease time.

```
module: ietf-dhcp
   +--rw dchp
      +--rw client
         +--rw interfaces* [interface]
            +--rw interface?   if:interface-ref
            +--rw client-id?   string
            +--rw lease?       uint32
```

DHCP State

The "dhcp-state" records the package statistic information and host allocated status, including server, relay and client.

```
module: ietf-dhcp
+--ro dhcp-state
         +--ro server {server}?
         |  +--ro packet-statistics
         |  |  +--ro interface?   if:interface-state-ref
         |  |  +--ro receive
         |  |  |  +--...      //The packet which server received from client
         |  |  +--ro send
         |  |     +--...      //The packet which server sends to client
         |  +--ro host
         |  |  +--ro interface?                string
         |  |  +--ro host-ip?                  string
         |  |  +--ro host-hardware-address?    string
         |  |  +--ro lease?                    uint32
         |  |  +--ro type?                     allocate-type
         |  +--ro ip-pool* [ip-pool-name]
         |     +--ro ip-pool-name          string
         |     +--ro gateway-ip?           inet:ip-address
         |     +--ro gateway-mask?         inet:ip-prefix
         |     +--ro used-ip-count?        uint32
         |     +--ro idle-ip-count?        uint32
         |     +--ro conflict-ip-count?    uint32
         |     +--ro total-ip-count?       uint32
         +--ro relay {relay}?
         |  +--ro packet-statistics
         |     +--ro interface?   if:interface-state-ref
         |     +--ro receive
         |     |  +--...     //The packet which are receives from client and server
         |     +--ro send
         |        +--...     //The packet which are sent to client and server
         +--ro client {client}?
            +--ro packet-statistics
               +--ro interface?   if:interface-state-ref
               +--ro receive
               |  +--...     //The packet which client received from server
               +--ro send
                  +--...     //The packet which client sends to server
```

Packet Definition

```
module: ietf-dhcp
+--ro dhcp-state
      +--ro packet-statistics
         +--ro client-packet  //The packet which client sends to server
         |  +--ro decline-packet?    uint32
         |  +--ro discover-packet?   uint32
         |  +--ro request-packet?    uint32
         |  +--ro release-packet?    uint32
         |  +--ro inform-packet?     Uint32
         +--ro server-packet  //The packet which server sends to client
            +--ro offer-packet?   uint32
            +--ro ack-packet?     uint32
            +--ro nack-packet?    uint32
```

DHCP RPC

The " dhcp-rpc" is designed to clean the packet statistics on server/relay/client node.

```
rpcs:
   +---x clean-server-statistics     {dhcp-server}?
   |  +---w input
   |  |  +---w interface?   -> /dhcp-state/server/packet-statistics/interface
   |  |  +---w clean-at?    yang:date-and-time
   |  +--ro output
   |     +--ro clean-finished-at?   yang:date-and-time
   +---x clean-relay-statistics      {dhcp-relay}?
   |  +---w input
   |  |  +---w interface?   -> /dhcp-state/relay/packet-statistics/interface
   |  |  +---w clean-at?    yang:date-and-time
   |  +--ro output
   |     +--ro clean-finished-at?   yang:date-and-time
   +---x clean-client-statistics     {dhcp-client}?
      +---w input
      |  +---w interface?   -> /dhcp-state/client/packet-statistics/interface
      |  +---w clean-at?    yang:date-and-time
      +--ro output
         +--ro clean-finished-at?   yang:date-and-time
```

Open Issues

- Keep alignment with DHCPv6 YANG data modeling.
- Regarding action, use RPC or Action(YANG1.1), need confirm with WG.
- On DHCP notification, including IP change, lease time expiration and so on, the related data model is not defined.
- How to handle the evolution of DHCP, new message and configuration item.

Next Steps

- Fix open issues
- Solicit comments
- WG consensus