DHC Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 28, 2016

Y. Cui
H. Wang
L. Sun
Tsinghua University
T. Lemon
Nominum
I. Farrer
Deutsche Telekom AG
September 25, 2015

YANG Data Model for DHCPv6 Configuration
draft-cui-dhc-dhcpv6-yang-04

Abstract

There has no unified method to configure DHCPv6 server ,relay and client itself, always pre-configured manually by operators.

IETF netmod WG has developed a general data model for NETCONF protocol, YANG data model [RFC6020].

This document defines a YANG data model for the configuration and management of DHCPv6 server, DHCPv6 relay and DHCPv6 client. With this model, the operators can configure and manage the devices by using NETCONF.

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 [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 March 28, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
  1.1. Terminology 3
2. Objectives 3
  2.1. DHCPv6 server 4
  2.2. DHCPv6 relay 4
  2.3. DHCPv6 client 4
3. DHCPv6 Tree Diagrams 4
  3.1. DHCPv6 Server Tree Diagrams 4
  3.2. DHCPv6 Relay Tree Diagrams 12
  3.3. DHCPv6 Client Tree Diagrams 14
  3.4. Notifications Mechanism for DHCPv6 19
4. DHCPv6 YANG Model 21
5. Security Considerations (TBD) 77
6. IANA Considerations (TBD) 77
7. Acknowledgements 77
8. References 77
  8.1. Normative References 77
  8.2. Informative References 78
Authors' Addresses 80

1. Introduction

This document defines a YANG data model for the configuration and management of DHCPv6 server, DHCPv6 relay and DHCPv6 client. With this model, the operators can configure and manage the devices by using NETCONF.

Model include three sub-modules:

- o DHCPv6 server
- o DHCPv6 relay
- o DHCPv6 client

For DHCPv6 client configuration, it is worth noting that as DHCPv6 itself a device configuration protocol, the intention of this document is not to replace client configuration of DHCPv6 options and parameters over the DHCPv6 protocol with the configuration of DHCPv6 options and parameters over NETCONF/YANG. The DHCPv6 client model is intended for the configuration of the DHCPv6 client function and also for obtaining read-only state data from the client which has been learnt via the normal DHCPv6 message flow. This gives an operator a better method for managing DHCPv6 clients and simplifies troubleshooting.

1.1. Terminology

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 [RFC2119].

The reader should be familiar with the terms defined in DHCPv6 [RFC3315] and relevant documents.

DHCPv6 tree diagrams provide a concise representation of a YANG module to help readers understand the module structure. The meaning if the symbols in these diagrams is as follows:

- o Brackets "[" and "]" enclose list keys.
- o Braces "{" and "}" enclose feature content.
- o Parentheses "(" and ")" enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Symbols after data node names: "?" means an optional node, and "*" denotes a list and leaf-list.
- o Abbreviations before data node names: "rw" means configuration data (read-write), and "ro" means state data (read-only).

2. Objectives

This document defines a YANG data model that can be used to configure and manage DHCPv6 server, DHCPv6 relay and DHCPv6 client.

2.1. DHCPv6 server

DHCPv6 server parameters.

2.2. DHCPv6 relay

DHCPv6 relay parameters.

2.3. DHCPv6 client

DHCPv6 client parameters.

3. DHCPv6 Tree Diagrams

3.1. DHCPv6 Server Tree Diagrams

```
+--rw dhcpv6
   +--rw server                                {dhcpv6-server}?
      +--rw serv-attributes
      |  +--rw name                            string
      |  +--rw duid                            duidtype
      |  +--rw enable                          boolean
      |  +--rw ipv6-address                    inet:ipv6-address
      |  +--rw description?                    string
      |  +--rw pd-function                     boolean
      |  +--rw stateless-service               boolean
      |  +--rw rapid-commit                    boolean
      |  +--rw store-client-link-layer?        boolean
      |  +--rw vendor-info
      |     +--rw ent-num                      uint32
      |     +--rw data*                        string
      +--rw option-sets
      |  +--rw option-set* [option-set-id]
      |     +--rw option-set-id                uint8
      |     +--rw new-option* [option-code]
      |     |  +--rw option-code               uint16
      |     |  +--rw option-name               string
      |     |  +--rw option-description        string
      |     |  +--rw option-reference?         string
      |     |  +--rw option-value              string
      |     +--rw user-class-value?            string
      |     +--rw enterprise-number?           uint32
      |     +--rw store-client-link-layer?     boolean
      |     +--rw preference-option
      |     |  +--rw enable                    boolean
      |     |  +--rw preference-value          uint8
      |     +--rw sip-server-option
      |     |  +--rw enable                    boolean
```

```
|     |  +--rw sip-server* [sip-serv-id]
|     |     +--rw sip-serv-id            uint8
|     |     +--rw sip-serv-domain-name   string
|     |     +--rw sip-serv-addr          inet:ipv6-address
|     +--rw dns-config-option
|     |  +--rw enable                    boolean
|     |  +--rw dns-server* [dns-serv-id]
|     |  |  +--rw dns-serv-id            uint8
|     |  |  +--rw dns-serv-addr          inet:ipv6-address
|     |  +--rw domain-search-list        string
|     +--rw nis-config-option
|     |  +--rw enable                    boolean
|     |  +--rw nis-server* [nis-serv-id]
|     |     +--rw nis-serv-id            uint8
|     |     +--rw nis-serv-addr          inet:ipv6-address
|     +--rw nis-plus-config-option
|     |  +--rw enable                    boolean
|     |  +--rw nis-plus-server* [nis-plus-serv-id]
|     |     +--rw nis-plus-serv-id       uint8
|     |     +--rw nis-plus-serv-addr     inet:ipv6-address
|     +--rw info-refresh-time-option
|     |  +--rw enable                    boolean
|     |  +--rw info-refresh-time         yang:timeticks
|     +--rw cli-fqdn-option
|     |  +--rw enable                    boolean
|     |  +--rw server-initiate-update    boolean
|     |  +--rw client-initiate-update    boolean
|     |  +--rw modify-name-from-cli      boolean
|     +--rw timezone-option
|     |  +--rw enable                    boolean
|     |  +--rw tz-posix                  string
|     |  +--rw tz-database               string
|     +--rw ntp-server-option
|     |  +--rw enable                    boolean
|     |  +--rw ntp-server* [ntp-serv-id]
|     |     +--rw ntp-serv-id            uint8
|     |     +--rw ntp-serv-addr          inet:ipv6-address
|     |     +--rw ntp-serv-mul-addr      inet:ipv6-address
|     |     +--rw ntp-serv-fqdn          string
|     +--rw sntp-server-option
|     |  +--rw enable                    boolean
|     |  +--rw sntp-server* [sntp-serv-id]
|     |     +--rw sntp-serv-id           uint8
|     |     +--rw sntp-serv-addr         inet:ipv6-address
|     +--rw network-boot-option
|     |  +--rw enable                    boolean
|     |  +--rw boot-file* [boot-file-id]
|     |     +--rw boot-file-id           uint8
```

```
            |     |     +--rw suitable-arch-type*    uint16
            |     |     +--rw suitable-net-if*       uint32
            |     |     +--rw boot-file-url          string
            |     |     +--rw boot-file-paras* [para-id]
            |     |        +--rw para-id             uint8
            |     |        +--rw parameter           string
            |     +--rw dslite-option
            |     |  +--rw enable                    boolean
            |     |  +--rw dslite-aftr-name          string
            |     +--rw kerberos-option
            |     |  +--rw enable                    boolean
            |     |  +--rw default-realm-name        string
            |     |  +--rw kdc-info* [kdc-id]
            |     |     +--rw kdc-id                 uint8
            |     |     +--rw priority               uint16
            |     |     +--rw weight                 uint16
            |     |     +--rw transport-type         uint8
            |     |     +--rw port-number            uint16
            |     |     +--rw kdc-ipv6-addr          inet:ipv6-address
            |     |     +--rw realm-name             string
            |     +--rw addr-selection-option
            |     |  +--rw enable                    boolean
            |     |  +--rw a-bit-set                 boolean
            |     |  +--rw p-bit-set                 boolean
            |     |  +--rw policy-table* [policy-id]
            |     |     +--rw policy-id              uint8
            |     |     +--rw label                  uint8
            |     |     +--rw precedence             uint8
            |     |     +--rw prefix-len             uint8
            |     |     +--rw prefix                 inet:ipv6-prefix
            |     +--rw sol-max-rt-option
            |     |  +--rw enable                    boolean
            |     |  +--rw sol-max-rt-value          yang:timeticks
            |     +--rw inf-max-rt-option
            |     |  +--rw enable                    boolean
            |     |  +--rw inf-max-rt-value          yang:timeticks
            |     +--rw pcp-server-option
            |     |  +--rw enable                    boolean
            |     |  +--rw pcp-server* [pcp-serv-id]
            |     |     +--rw pcp-serv-id            uint8
            |     |     +--rw pcp-serv-addr*         inet:ipv6-address
            |     +--rw s46-rule-option
            |     |  +--rw enable                    boolean
            |     |  +--rw s46-rule* [rule-id]
            |     |     +--rw rule-id                uint8
            |     |     +--rw rule-type              enumeration
            |     |     +--rw ea-len                 uint8
            |     |     +--rw prefix4-len            uint8
```

```
            |     |     +--rw ipv4-prefix            inet:ipv4-prefix
            |     |     +--rw prefix6-len            uint8
            |     |     +--rw ipv6-prefix            inet:ipv6-prefix
            |     |     +--rw port-parameter
            |     |        +--rw offset              uint8
            |     |        +--rw psid-len            uint8
            |     |        +--rw psid                uint16
            |     +--rw s46-br-option
            |     |  +--rw enable                    boolean
            |     |  +--rw br* [br-id]
            |     |      +--rw br-id                 uint8
            |     |      +--rw br-ipv6-addr          inet:ipv6-address
            |     +--rw s46-dmr-option
            |     |  +--rw enable                    boolean
            |     |  +--rw dmr* [dmr-id]
            |     |     +--rw dmr-id                 uint8
            |     |     +--rw dmr-prefix6-len        uint8
            |     |     +--rw dmr-ipv6-prefix        inet:ipv6-prefix
            |     +--rw s46-v4-v6-binding-option
            |        +--rw enable                    boolean
            |        +--rw ce* [ce-id]
            |           +--rw ce-id                  uint8
            |           +--rw ipv4-addr              inet:ipv4-address
            |           +--rw bind-prefix6-len       uint8
            |           +--rw bind-ipv6-prefix       inet:ipv6-prefix
            |           +--rw port-parameter
            |              +--rw offset              uint8
            |              +--rw psid-len            uint8
            |              +--rw psid                uint16
            +--rw network-ranges
            |  +--rw option-set [option-set-id]
            |  +--rw network-range* [network-range-id]
            |     +--rw network-range-id             uint8
            |     +--rw network-description          string
            |     +--rw network-prefix               inet:ipv6-prefix
            |     +--rw inherit-option-set           boolean
            |     +--rw option-set [option-set-id]
            |     +--rw address-pools
            |     |  +--rw address-pool* [pool-id]
            |     |  |  +--rw pool-id                uint8
            |     |  |  +--rw pool-prefix            inet:ipv6-prefix
            |     |  |  +--rw start-address          inet:ipv6-address
            |     |  |  +--rw end-address            inet:ipv6-address
            |     |  |  +--rw preferred-lifetime     yang:timeticks
            |     |  |  +--rw valid-lifetime         yang:timeticks
            |     |  |  +--ro total-ipv6-count       uint64
            |     |  |  +--ro used-ipv6-count        uint64
            |     |  |  +--rw utilization-ratio      threshold
```

```
            |     |  |  +--rw inherit-option-set     boolean
            |     |  |  +--rw option-set [option-set-id]
            |     |  |  +--rw reserved-addresses
            |     |  |     +--rw static-binding* [cli-id]
            |     |  |     |  +--rw cli-id           uint32
            |     |  |     |  +--rw duid             duidtype
            |     |  |     |  +--rw reserv-addr*     inet:ipv6-address
            |     |  |     +--rw other-reserv-addr*  inet:ipv6-address
            |     |  +--ro binding-info* [cli-id]
            |     |     +--ro cli-id                uint32
            |     |     +--ro duid                  duidtype
            |     |     +--ro cli-ia* [iaid]
            |     |        +--ro ia-type            string
            |     |        +--ro iaid               uint32
            |     |        +--ro cli-addr*          inet:ipv6-address
            |     |        +--ro pool-id?           uint8
            |     +--rw prefix-pools
            |     |  +--rw prefix-pool* [pool-id]
            |     |  |  +--rw pool-id               uint8
            |     |  |  +--rw prefix                inet:ipv6-prefix
            |     |  |  +--rw prefix-length         uint8
            |     |  |  +--rw preferred-lifetime    yang:timeticks
            |     |  |  +--rw valid-lifetime        yang:timeticks
            |     |  |  +--rw utilization-ratio     threshold
            |     |  |  +--rw inherit-option-set    boolean
            |     |  |  +--rw option-set [option-set-id]
            |     |  |  +--rw reserved-prefixes
            |     |  |     +--rw static-binding* [cli-id]
            |     |  |     |  +--rw cli-id           uint32
            |     |  |     |  +--rw duid             duidtype
            |     |  |     |  +--rw reserv-prefix-len  uint8
            |     |  |     |  +--rw reserv-prefix    inet:ipv6-prefix
            |     |  |     +--rw exclude-prefix-len  uint8
            |     |  |     +--rw exclude-prefix      inet:ipv6-prefix
            |     |  |     +--rw other-reserv-prefix*  [reserv-id]
            |     |  |        +--rw reserv-id        uint8
            |     |  |        +--rw prefix-len       uint8
            |     |  |        +--rw prefix           inet:ipv6-prefix
            |     |  +--ro binding-info* [cli-id]
            |     |     +--ro cli-id                uint32
            |     |     +--ro duid                  duidtype
            |     |     +--ro cli-iapd* [iaid]
            |     |        +--ro iaid               uint32
            |     |        +--ro cli-prefix*        uint32
            |     |        +--ro cli-prefix-len*    uint8
            |     |        +--ro pool-id?           uint8
            |     +--rw hosts
            |        +--rw host* [cli-id]
```

```
          |           +--rw cli-id                 uint32
          |           +--rw duid                   duidtype
          |           +--rw inherit-option-set     boolean
          |           +--rw option-set [option-set-id]
          |           +--rw nis-domain-name?       string
          |           +--rw nis-plus-domain-name?  string
          +--rw relay-opaque-paras
          |  +--rw relays* [relay-name]
          |     +--rw relay-name                   string
          |     +--rw interface-info* [if-name]
          |     |  +--rw if-name                   string
          |     |  +--rw interface-id              string
          |     +--rw subscribers* [subscriber]
          |     |  +--rw subscriber                uint8
          |     |  +--rw subscriber-id             string
          |     +--rw remote-host* [ent-num]
          |        +--rw ent-num                   uint32
          |        +--rw remote-id                 string
          +--rw rsoo-enabled-options
          |  +--rw rsoo-enabled-option* [option-code]
          |     +--rw option-code                  uint16
          |     +--rw description                  string
          +--ro packet-stats
             +--ro solicit-count                   uint32
             +--ro request-count                   uint32
             +--ro renew-count                     uint32
             +--ro rebind-count                    uint32
             +--ro decline-count                   uint32
             +--ro release-count                   uint32
             +--ro info-req-count                  uint32
             +--ro advertise-count                 uint32
             +--ro confirm-count                   uint32
             +--ro reply-count                     uint32
             +--ro reconfigure-count               uint32
             +--ro relay-forward-count             uint32
             +--ro relay-reply-count               uint32
```

Figure 1: DHCPv6 Data Model Structure

Introduction of important nodes:

o serv-attributes: This container contains basic attributes of a DHCPv6 server such as DUID, server name and so on. Some optional functions that can be provided by the server is also included.

o duid: Each server and client has only one DUID (DHCP Unique Identifier). The DUID here identifies a unique DHCPv6 server for

clients. DUID consists of a two-octet type field and an arbitrary length (no more than 128 bytes) content field.

- o pd-function: Whether the server can act as a delegating router to perform prefix delegation ([RFC3633]).

- o stateless-service: A boolean value specifies whether the server support client-server exchanges involving two messages defined in ([RFC3315]).

- o rapid-commit: Setting the value to '1' represents the server support the Solicit-Reply message exchange. '0' means the server will simply ignore the Rapid Commit option in Solicit message.

- o option-sets: DHCPv6 employs various options to carry additional information and parameters in DHCP messages. This container defines all the possible options that need to be configured at the server side. The relevant RFCs that define those options include: [RFC3315], [RFC3319], [RFC3646], [RFC3898], [RFC4242], [RFC4704], [RFC4833], [RFC5908], [RFC5970], [RFC4075], [RFC6334], [RFC6784], [RFC7078], [RFC7083], [RFC7291], [RFC7598].

- o option-set: A server may allow different option sets to be configured for different conditions (i.e. different networks, clients and etc). This "option-set" list enables various sets of options being defined and configured in a single server. Different sets are distinguished by the key called "option-set-id". All the possible options discussed above are defined in the list and each option is corresponding to a container. Since all the options in the list are optional, each container in this list has a boolean parameter called "enable" to indicate whether this option (container) will be included in the current option set or not. With the "new-option" container, it is easy to extend the model when new options are defined.

- o network-ranges: This model supports a hierarchy to achieve dynamic configuration. That is to say we could configure the server at different levels through this model. The top level is a global level which is defined as the container "network-ranges". The following levels are defined as sub-containers under it. The "network-ranges" contains the parameters (e.g. option-sets) that would be allocated to all the clients served by this server.

- o network-range: Under the "network-ranges" container, a "network-range" list is defined to configure the server at a network level which is also considered as the second level. Different network are identified by the key "network-range-id". This is because a

server may have different configuration parameters (e.g. option sets) for different networks.

- o address-pools: Under the "network-range" list, a container describes the DHCPv6 server's address pools for a specific network is defined. This container supports the server to be configured at a pool level.

- o address-pool: A DHCPv6 server can be configured with several address pools for a specific network. This list defines such address pools which are distinguish by the key called "pool-id".

- o binding-info: A list records a binding information for each DHCPv6 client that has already been allocated IPv6 addresses.

- o prefix-pools: If a server supports prefix delegation function, this container under the "network-range" list will be valid to define the delegating router's prefix pools for a specific network. This container also supports the server to be configured at a pool level.

- o prefix-pool: Similar to server's address pools, a delegating router can also be configured with multiple prefix pools specified by a list called "prefix-pool".

- o binding-info: A list records a binding information for each DHCPv6 requesting router that has already been configured IPv6 prefixes.

- o hosts: A server may also desire to be configured at a host level under some circumstances. This container include a list called "host" to allow the server carrying different parameters (e.g. option sets) for different hosts.

- o relay-opaque-paras: This container contains some opaque values in Relay Agent options that need to be configured on the server side only for value match. Such Relay Agent options include Interface-Id option, Remote-Id option and Subscriber-Id option.

- o rsoo-enabled-options: [RFC6422] requires that the server SHOULD have an administrator-configurable list of RSOO-enabled options. This container include a list called "rsoo-enabled-option" to allow new RSOO-enabled options to be defined at the server side.

- o packet-stats: A container presents the packet statistics related to the DHCPv6 server.

3.2. DHCPv6 Relay Tree Diagrams

```
+--rw dhcpv6
   +-- ...
   |
   +--rw relay                                  {dhcpv6-relay}?
      +--rw relay-attributes
      |  +--rw name                             string
      |  +--rw enable                           boolean
      |  +--rw ipv6-address                     inet:ipv6-address
      |  +--rw description?                     string
      |  +--rw dest-addrs*                      inet:ipv6-address
      |  +--rw subscribers* [subscriber]
      |  |  +--rw subscriber                    uint8
      |  |  +--rw subscriber-id                 string
      |  +--rw remote-host* [entNum]
      |  |  +--rw ent-num                       uint32
      |  |  +--rw remote-id                     string
      |  +--rw vendor-info
      |     +--rw ent-num                       uint32
      |     +--rw data*                         string
      +--rw relay-supplied-options-option
      |  +--rw rsoo-set* [rsoo-set-id]
      |     +--rw rsoo-set-id               uint8
      |     +--rw erp-local-domain-name-option
      |        +--rw enable                 boolean
      |        +--rw erp-for-client* [cli-id]
      |           +--rw cli-id              uint32
      |           +--rw duid                duidtype
      |           +--rw erp-name            string
      +--rw relay-interfaces
      |  +--rw relay-if* [if-name]
      |     +--rw if-name                   string
      |     +--rw enable                    boolean
      |     +--rw interface-id?             string
      |     +--rw rsoo-set [rsoo-set-id]
      |     +--rw pd-route* [pd-route-id]
      |     |  +--rw pd-route-id            uint8
      |     |  +--rw requesting-router-id   uint32
      |     |  +--rw delegating-router-id   uint32
      |     |  +--rw next-router            inet:ipv6-address
      |     |  +--rw last-router            inet:ipv6-address
      |     +--rw next-entity* [dest-addr]
      |        +--rw dest-addr              inet:ipv6-address
      |        +--rw available              boolean
      |        +--rw multicast              boolean
      |        +--rw server                 boolean
      |        +--ro packet-stats
```

```
        |           +--ro cli-packet-rvd-count  uint32
        |           +--ro solicit-rvd-count     uint32
        |           +--ro request-rvd-count     uint32
        |           +--ro renew-rvd-count       uint32
        |           +--ro rebind-rvd-count      uint32
        |           +--ro decline-rvd-count     uint32
        |           +--ro release-rvd-count     uint32
        |           +--ro info-req-rvd-count    uint32
        |           +--ro relay-for-rvd-count   uint32
        |           +--ro relay-rep-rvd-count   uint32
        |           +--ro packet-to-cli-count   uint32
        |           +--ro adver-sent-count      uint32
        |           +--ro confirm-sent-count    uint32
        |           +--ro reply-sent-count      uint32
        |           +--ro reconfig-sent-count   uint32
        |           +--ro relay-for-sent-count  uint32
        |           +--ro relay-rep-sent-count  uint32
        +--ro relay-stats
           +--ro cli-packet-rvd-count            uint32
           +--ro relay-for-rvd-count             uint32
           +--ro relay-rep-rvd-count             uint32
           +--ro packet-to-cli-count             uint32
           +--ro relay-for-sent-count            uint32
           +--ro relay-rep-sent-count            uint32
           +--ro discarded-packet-count          uint32
```

Introduction of important nodes:

o relay-attributes: A container describes some basic attributes of the relay agent including some relay agent specific options data that need to be configured previously. Such options include Remote-Id option and Subscriber-Id option.

o dest-addrs: Each DHCPv6 relay agent may be configured with a list of destination addresses. This node defines such a list of IPv6 addresses that may include unicast addresses, multicast addresses or other addresses.

o relay-supplied-options-option: DHCPv6 relay agent could provide some information that would be useful to DHCPv6 client. Since relay agent cannot provide options directly to the client, [RFC6422] defines RSOO-enabled options to propose options for the server to send to the client. This container modelled such RSOO-enabled options.

o rsoo-set: This list under the "relay-supplied-options-option" container is similar to the "option-set" defined in server feature. It allows the relay to implement several sets of RSOO-

enabled options for different interfaces. The list only include the EAP Re-authentication Protocol (ERP) Local Domain Name DHCPv6 Option defined in [RFC6440], since it is the only one RSOO-enabled options accepted by IANA so far.

- o relay-interfaces: The "relay-interfaces" defines common configuration and state parameters of the interfaces belonging to a DHCPv6 relay agent.

- o relay-if: A list under "relay-interfaces" container that describes a specific interface and its corresponding parameters. Here we use a string called "if-name" as the key of list.

- o pd-route: A sub-container of "relay-if" which describes the route for delegated prefixes into the provider edge router.

- o next-entity: This node defines a list that is used to describe the next hop entity of this relay agent. Different entities are distinguished by their addresses.

- o packet-stats: A container shows packet state information of a specific data communication.

- o relay-stats: The "relay-stats" container records and presents the overall packet statistics of the relay agent.

3.3. DHCPv6 Client Tree Diagrams

```
+--rw dhcpv6
   +-- ...
   |
   +--rw client                                {dhcpv6-client}?
      +--rw client-interfaces
         +--rw client-if* [if-name]
            +--rw if-name                     string
            +--rw cli-id                      uint32
            +--rw duid                        duidtype
            +--rw enable                      boolean
            +--rw description?                string
            +--rw pd-function                 boolean
            +--rw rapid-commit                boolean
            +--rw mo-tab
            |  +--rw m-tab                    boolean
            |  +--rw o-tab                    boolean
            +--rw oro-options
            |  +--rw oro-option* [option-code]
            |     +--rw option-code           uint16
            |     +--rw description           string
```

```
+--rw client-configured-options
|  +--rw new-cli-option* [option-code]
|  |  +--rw option-code            uint16
|  |  +--rw option-name            string
|  |  +--rw option-description     string
|  |  +--rw option-reference?      string
|  |  +--rw option-value           string
|  +--rw user-class-option
|  |  +--rw enable                 boolean
|  |  +--rw user-class* [user-class-id]
|  |     +--rw user-class-id       uint8
|  |     +--rw user-class-info     string
|  +--rw vendor-class-option
|  |  +--rw enable                 boolean
|  |  +--rw ent-num                uint32
|  |  +--rw data*                  string
|  +--rw client-fqdn-option
|  |  +--rw enable                 boolean
|  |  +--rw fqdn                   string
|  |  +--rw server-initiate-update boolean
|  |  +--rw client-initiate-update boolean
|  +--rw client-architecture-type-option
|  |  +--rw enable                 boolean
|  |  +--rw architecture-types* [type-id]
|  |  |  +--rw type-id             uint16
|  |  |  +--rw most-preferred      boolean
|  +--rw client-network-interface-option
|  |  +--rw enable                 booelan
|  |  +--rw type                   uint8
|  |  +--rw major                  uint8
|  |  +--rw minor                  uint8
|  +--rw kerberos-principal-name-option
|  |  +--rw enable                 boolean
|  |  +--rw principal-name         string
|  +--rw client-link-layer-addr-option
|     +--rw enable                 boolean
|     +--rw link-layer-type        uint16
|     +--rw link-layer-addr        string
+--ro identity-associations
|  +--ro identity-association* [iaid]
|     +--ro iaid                   uint32
|     +--ro ia-type                string
|     +--ro ipv6-addr*             inet:ipv6-address
|     +--ro ipv6-prefix*           inet:ipv6-prefix
|     +--ro prefix-length*         uint8
|     +--ro t1-time                yang:date-and-time
|     +--ro t2-time                yang:date-and-time
|     +--ro preferred-lifetime     yang:timeticks
```

```
|     +--ro valid-lifetime         yang:timeticks
+--ro if-other-paras
|  +--ro uni-dhcpv6-serv-addr      inet:ipv6-address
|  +--ro dns-paras
|  |  +--ro domain-search-list     string
|  |  +--ro dns-servers* [dns-serv-id]
|  |     +--ro dns-serv-id         uint8
|  |     +--ro dns-serv-addr       inet:ipv6-address
|  +--ro sip-paras
|  |  +--ro sip-servers* [sip-serv-id]
|  |     +--ro sip-serv-id         uint8
|  |     +--ro sip-serv-addr       inet:ipv6-address
|  |     +--ro sip-serv-domain-name  string
|  +--ro nis-paras
|  |  +--ro nis-domain-name        string
|  |  +--ro nis-server* [nis-serv-id]
|  |     +--ro nis-serv-id         uint8
|  |     +--ro nis-serv-addr       inet:ipv6-address
|  +--ro nis-plus-paras
|  |  +--ro nis-plus-domain-name   string
|  |  +--ro nis-plus-server* [nis-plus-serv-id]
|  |     +--ro nis-plus-serv-id    uint8
|  |     +--ro nis-plus-serv-addr  inet:ipv6-address
|  +--ro info-refresh-time         yang:timeticks
|  +--ro time-zone-paras
|  |  +--ro tz-posix               string
|  |  +--ro tz-database            string
|  +--ro cli-fqdn                  string
|  +--ro ntp-paras
|  |  +--ro ntp-server* [ntp-serv-id]
|  |     +--ro ntp-serv-id         uint8
|  |     +--ro ntp-serv-addr       inet:ipv6-address
|  |     +--ro ntp-serv-mul-addr   inet:ipv6-address
|  |     +--ro ntp-serv-fqdn       string
|  +--ro sntp-paras
|  |  +--ro sntp-server* [sntp-serv-id]
|  |     +--ro sntp-serv-id         uint8
|  |     +--ro sntp-serv-addr       inet:ipv6-address
|  +--ro network-boot-paras
|  |  +--ro boot-file* [boot-file-id]
|  |     +--ro boot-file-id           uint8
|  |     +--ro suitable-arch-type*    uint16
|  |     +--ro suitable-net-if*       uint32
|  |     +--ro boot-file-url          string
|  |     +--ro boot-file-paras* [para-id]
|  |        +--ro para-id             uint8
|  |        +--ro parameter           string
|  +--ro kerberos-paras
```

```
                |  |  +--ro default-realm-name        string
                |  |  +--ro kdc-info* [kdc-id]
                |  |     +--ro kdc-id                 uint8
                |  |     +--ro priority               uint16
                |  |     +--ro weight                 uint16
                |  |     +--ro transport-type         uint8
                |  |     +--ro port-number            uint16
                |  |     +--ro kdc-ipv6-addr          inet:ipv6-address
                |  |     +--ro realm-name             string
                |  +--ro addr-selection-paras
                |  |  +--ro automatic-row-add         boolean
                |  |  +--ro prefer-temporary-addr     boolean
                |  |  +--ro policy-table* [policy-id]
                |  |     +--ro policy-id              uint8
                |  |     +--ro label                  uint8
                |  |     +--ro precedence             uint8
                |  |     +--ro prefix-len             uint8
                |  |     +--ro prefix                 inet:ipv6-prefix
                |  +--ro sol-max-rt                   yang:timeticks
                |  +--ro inf-max-rt                   yang:timeticks
                |  +--ro pcp-paras
                |  |  +--ro pcp-server* [pcp-serv-id]
                |  |     +--ro pcp-serv-id            uint8
                |  |     +--ro pcp-serv-addr          inet:ipv6-address
                |  +--ro s46-rule-paras
                |  |  +--ro s46-rule* [rule-id]
                |  |     +--ro rule-id                uint8
                |  |     +--ro rule-type              enumeration
                |  |     +--ro ea-len                 uint8
                |  |     +--ro prefix4-len            uint8
                |  |     +--ro ipv4-prefix            inet:ipv4-prefix
                |  |     +--ro prefix6-len            uint8
                |  |     +--ro ipv6-prefix            inet:ipv6-prefix
                |  |     +--ro port-parameter
                |  |        +--ro offset              uint8
                |  |        +--ro psid-len            uint8
                |  |        +--ro psid                uint16
                |  +--ro s46-br-paras
                |  |  +--ro br* [br-id]
                |  |     +--ro br-id                  uint8
                |  |     +--ro br-ipv6-addr           inet:ipv6-address
                |  +--ro s46-dmr-paras
                |  |  +--ro dmr* [dmr-id]
                |  |     +--ro dmr-id                 uint8
                |  |     +--ro dmr-prefix6-len        uint8
                |  |     +--ro dmr-ipv6-prefix        inet:ipv6-prefix
                |  +--ro s46-v4-v6-binding-paras
                |  |  +--ro ipv4-addr                 inet:ipv6-address
```

```
               |  |  +--ro bind-prefix6-len          uint8
               |  |  +--ro port-parameter
               |  |     +--ro offset                 uint8
               |  |     +--ro psid-len               uint8
               |  |     +--ro psid                   uint16
               |  +--ro erp-local-domain-name        string
               +--ro supported-options
               |  +--ro supported-option* [option-code]
               |     +--ro option-code            uint16
               |     +--ro description            string
               +--ro packet-stats
                  +--ro solicit-count             uint32
                  +--ro request-count             uint32
                  +--ro renew-count               uint32
                  +--ro rebind-count              uint32
                  +--ro decline-count             uint32
                  +--ro release-count             uint32
                  +--ro info-req-count            uint32
                  +--ro advertise-count           uint32
                  +--ro confirm-count             uint32
                  +--ro reply-count               uint32
                  +--ro reconfigure-count         uint32
```

Introduction of important nodes:

- o client-interfaces: A client may have several interfaces, it is more reasonable to configure and manage parameters on the interface-level. This container includes configuration and state data of a DHCPv6 client in a per-interface manner.

- o client-if: The list defines a specific client interface and its data. Different interfaces are distinguished by the "ifName" key which is a configurable string value.

- o duid: Each server and client has only one DUID (DHCP Unique Identifier). The DUID here will be carried in the Client ID option to identify a specific DHCPv6 client. This leaf are same as the "duid" leaf in "dhcpv6-server" feature.

- o pd-function: Whether the client can act as a requesting router to request prefixes using prefix delegation ([RFC3633]).

- o rapid-commit: '1' indicates a client can initiate a Solicit-Reply message exchange by adding a Rapid Commit option in Solicit message. '0' means the client is not allowed to add a Rapid Commit option to request addresses in a two-message exchange pattern.

- o mo-tab: The management tab label indicates the operation mode of the DHCPv6 client. 'm'=1 and 'o'=1 indicate the client will use DHCPv6 to obtain all the configuration data. 'm'=1 and 'o'=0 are a meaningless combination. 'm'=0 and 'o'=1 indicate the client will use stateless DHCPv6 to obtain configuration data apart from addresses/prefixes data. 'm'=0 and 'o'=0 represent the client will not use DHCPv6 but use SLAAC to achieve configuration.

- o oro-options: This container provide a way to configure the list of options that the client will request in its ORO option.

- o client-configured-options: Similar to the server, the client also need to configure some options to fulfil some desired functions. This container include all the potential options that need to be configured at the client side. The relevant RFCs that define those options include: [RFC3315], [RFC4704], [RFC5970], [RFC6784], [RFC6939].

- o identity-association: IA is a construct through which a server and a client can identify, group, and manage a set of related IPv6 addresses. The key of the "identity-association" list is a 4-byte number IAID defined in [RFC3315] .

- o if-other-paras: A client can obtain extra configuration data other than address and prefix information through DHCPv6 options. This container describes such data the client was configured through DHCPv6. The potential configuration data may include DNS server parameters, SIP server parameters and etc.

- o supported-options: This state data container defines a list of options supported by the client for administrator to interrogate a client's capabilities.

- o packet-stats: A container records all the packet status information of a specific interface.

3.4. Notifications Mechanism for DHCPv6

```
+--rw dhcpv6
   +-- ...
   |
   +--n notifications
      +--n dhcpv6-server-event                {dhcpv6-server}?
      |  +--n pool-running-out
      |  |  +--ro utilization-ratio           uint16
      |  |  +--ro duid                        duidtype
      |  |  +--ro serv-name?                  string
      |  |  +--ro pool-name                   string
      |  +--n invalid-client-detected
      |     +--ro duid                        duidtype
      |     +--ro description?                string
      +--n dhcpv6-relay-event                 {dhcpv6-relay}?
      |  +--n topo-changed
      |     +--ro relay-if-name               string
      |     +--ro first-hop                   boolean
      |     +--ro last-entity-addr            inet:ipv6-address
      +--n dhcpv6-client-event                {dhcpv6-client}?
         +--n ia-lease-event
         |  +--ro event-type                  enumeration
         |  +--ro duid                        duidtype
         |  +--ro iaid                        uint32
         |  +--ro serv-name?                  string
         |  +--ro description?                string
         +--n invalid-ia-detected
         |  +--ro duid                        duidtype
         |  +--ro iaid                        uint32
         |  +--ro serv-name?                  string
         |  +--ro description?                string
         +--n retransmission-failed
         |  +--ro duid                        duidtype
         |  +--ro description                 enumeration
         +--n failed-status-turn-up
            +--ro duid                        duidtype
            +--ro status-code                 enumeration
```

Introduction of notifications:

o pool-running-out: raised when the address/prefix pool is going to run out. A threshold for utilization ratio of the pool has been defined in the server feature so that it will notify the administrator when the utilization ratio reaches the threshold, and such threshold is a settable parameter.

o invalid-client-detected: raised when the server has found a client which can be regarded as a potential attacker. Some description could also be included.

- o topo-changed: raised when the topology of the relay agent is changed.

- o ia-lease-event: raised when the client was allocated a new IA from the server or it renew/rebind/release its current IA.

- o invalid-ia-detected: raised when the identity association of the client can be proved to be invalid. Possible condition includes duplicated address, illegal address, etc.

- o retransmission-failed: raised when the retransmission mechanism defined in [RFC3315] is failed.

- o failed-status-turn-up: raised when the client receives a message includes an unsuccessful Status Code option.

4. DHCPv6 YANG Model

This module imports typedefs from [RFC6991].

```
<CODE BEGINS> file "ietf-dhcpv6@2015-09-25.yang"

module ietf-dhcpv6 {
    namespace "urn:ietf:params:xml:ns:yang:ietf-dhcpv6";
    prefix "dhcpv6";

    import ietf-inet-types {
        prefix inet;
        revision-date "2013-07-15";
    }
    import ietf-yang-types {
        prefix yang;
        revision-date "2013-07-15";
    }

    organization "dhc wg";
    contact "yong@csnet1.cs.tsinghua.edu.cn
            wangh13@mails.tsinghua.edu.cn
            lh.sunlinh@gmail.com
            Ted.Lemon@nominum.com
            ian.farrer@telekom.de";

    description "This model defines a YANG data model that can be
        used to configure and manage DHCPv6 server, DHCPv6 relay and
    DHCPv6 client.";

    revision 2015-09-25 {
        description "version04: Correct duid and grammar errors.";
```

```
        reference "I-D: draft-cui-dhc-dhcpv6-yang-04"
    }
    revision 2015-07-01 {
        description "version03: Correct grammar errors.";

        reference "I-D: draft-cui-dhc-dhcpv6-yang-03"
    }
    revision 2015-04-13 {
        description "version02: Correct grammar errors.";

        reference "I-D: draft-cui-dhc-dhcpv6-yang-02"
    }
    revision 2015-04-02 {
        description "version01: Correct grammar errors, Reuse
            groupings, Update 'dhcpv6-realy' feature, Add
                notifications.";

        reference "I-D: draft-cui-dhc-dhcpv6-yang-01"
    }
    revision 2015-03-04 {
        description "version00: Initial revision.";

        reference "I-D: draft-cui-dhc-dhcpv6-yang-00"
    }


/*
 * Features
 */

    feature dhcpv6-server {
        description
            "Server in DHCPv6.";
        reference
            "RFC3315";
    }

    feature dhcpv6-relay {
        description
            "Relay agent in DHCPv6.";
        reference
            "RFC3315";
    }

    feature dhcpv6-client {
        description
             "Client in DHCPv6.";
        reference
```

```
                "RFC3315";
    }
/*
 * Typedef
 */

    typedef duidtype {
            type union {
                    type uint16 {
                            description
                                "This uint16 specifies the type of this duid";
                        }
                        type string {
                            pattern '(([0-9a-fA-F]{2}){2,128})';
                            description
                "This should be a variable length value
                no more than 128 octets."
                        }
                }
        }

    typedef threshold {
        description "Threshold value in percent";
        type union {
            type uint16 {
                range 0..100;
                description "threshold value";
            }
            type enumeration {
                enum "disabled" {
                    description "No threshold";
                }
            }
        }
    }
/*
 * Grouping
 */

    grouping vendor-infor {
        container vendor-info {
            description "";
            leaf ent-num {
                type uint32;
                mandatory true;
                description "enterprise number";
            }
            leaf-list data {
```

```
                type string;
                description "specific vendor info";
            }
        }
    }

    grouping portset-para {
        container port-parameter {
            leaf offset {
                type uint8;
                mandatory true;
                description "offset in a port set";
            }
            leaf psid-len {
                type uint8;
                mandatory true;
                description "length of a psid";
            }
            leaf psid {
                type uint16;
                mandatory true;
                description "psid value";
            }
        }
    }

/*
 * Data Nodes
 */

    container server {
        if-feature dhcpv6-server;
         description "server portion";
        container serv-attributes {
            description "This container contains basic attributes
            of a DHCPv6 server such as DUID, server name and so
            on. Some optional functions that can be provided by
            the server is also included.";
            leaf name {
                type string;
                mandatory true;
                description "server's name";
            }
            leaf duid {
                            type duidtype;
                mandatory true;
                description "DHCP Unique Identifer";
                        }
```

```
            leaf enable {
                type boolean;
                mandatory true;
                description "whether to enable the server";
            }
            leaf ipv6-address {
                type inet:ipv6-address;
                mandatory true;
                description "server's IPv6 address";
            }
            leaf description {
                type string;
                description "description of the server";
            }
            leaf pd-function {
                type boolean;
                mandatory true;
                description "Whether the server can act as a
                delegating router to perform prefix delegation
                                ([RFC3633]).";
            }
            leaf stateless-service {
                type boolean;
                mandatory true;
                description "A boolean value specifies whether
                the server support client-server exchanges
                involving two messages defined in ([RFC3315]).";
            }
            leaf rapid-commit {
                type boolean;
                mandatory true;
                description "A boolean value specifies whether
                the server support client-server exchanges
                involving two messages defined in ([RFC3315]).";
            }
            leaf store-client-link-layer {
                type boolean;
                description "whether to store the clients'
                link-layer addresses";
            }
            uses vendor-infor;
            }
        container option-sets {
            description "*";
            list option-set {
                key option-set-id;
                description "*";
                leaf option-set-id {
```

```
                    type uint8;
                    mandatory true;
                    description "the option-set-id key";
                }
                                list new-option {
                    key option-code;
                    description "*";
                    leaf option-code {
                        type uint16;
                        manadatory true;
                        description "the option code key";
                    }
                    leaf option-name {
                        type string;
                        manadatory true;
                        description "the new option's name";
                    }
                    leaf option-description {
                        type string;
                        manadatory true;
                        description "description of new option";
                    }
                    leaf option-reference {
                        type string;
                        description "reference to the
                        specification";
                    }
                    leaf option-value {
                        type string;
                        manadatory true;
                        description "the new option's value";
                    }
                }
                leaf user-class-value {
                    type string;
                    description "use class option's value";
                }
                leaf enterprise-number {
                    type uint32;
                    description "*";
                }
                leaf store-client-link-layer {
                    type boolean;
                    description "*";
                }
                container preference-option {
                    description "*";
                    leaf enable {
```

```
                    type boolean;
                    mandatory true;
                    description "*";
                }
                leaf preference-value {
                    type uint8;
                    mandatory true;
                    description "*";
                }
            }
            container sip-server-option {
                description "*";
                leaf enable {
                    type boolean;
                    mandatory true;
                    description "*";
                }
                list sip-server {
                    key sip-serv-id;
                    leaf sip-serv-id {
                        type uint8;
                        mandatory true;
                        description "*";
                    }
                    leaf sip-serv-domain-name {
                        type string;
                        mandatory true;
                        description "*";
                    }
                    leaf sip-serv-addr {
                        type inet:ipv6-address;
                        mandatory true;
                        description "*";
                    }
                }
            }
            container dns-config-option {
                description "*";
                leaf enable {
                    type boolean;
                    mandatory true;
                    description "*";
                }
                list dns-server {
                    key dns-serv-id;
                    leaf dns-serv-id {
                        type uint8;
                        mandatory true;
```

```
                        description "*";
                    }
                    leaf dns-serv-addr {
                        type inet:ipv6-address;
                        mandatory true;
                        description "*";
                    }
                }
                leaf domain-search-list {
                    type string;
                    mandatory true;
                    description "*";
                }
            }
            container nis-config-option {
                description "*";
                leaf enable {
                    type boolean;
                    mandatory true;
                    description "*";
                }
                list nis-server {
                    key nis-serv-id;
                    description "*";
                    leaf nis-serv-id {
                        type uint8;
                        mandatory true;
                        description "*";
                    }
                    leaf nis-serv-addr {
                        type inet:ipv6-address;
                        mandatory true;
                        description "*";
                    }
                }
            }
            container nis-plus-config-option {
                description "*";
                leaf enable {
                    type boolean;
                    mandatory true;
                    description "*";
                }
                list nis-plus-server {
                    key nis-plus-serv-id;
                    description "*";
                    leaf nis-plus-serv-id {
                        type uint8;
```

```
                    mandatory true;
                    description "*";
                }
                leaf nis-plus-serv-addr {
                    type inet:ipv6-address;
                    mandatory true;
                    description "*";
                }
            }
        }
        container info-refresh-time-option {
            description "*";
            leaf enable {
                type boolean;
                mandatory true;
                description "*";
            }
            leaf info-refresh-time {
                type yang:timeticks;
                mandatory true;
                description "*";
            }
        }
        container cli-fqdn-option {
            description "*";
            leaf enable {
                type boolean;
                mandatory true;
                description "*";
            }
            leaf server-initiate-update {
                type boolean;
                mandatory true;
                description "*";
            }
            leaf client-initiate-update {
                type boolean;
                mandatory true;
                description "*";
            }
            leaf modify-name-from-cli {
                type boolean;
                mandatory true;
                description "*";
            }
        }
        container timezone-option {
            description "*";
```

```
                    leaf enable {
                        type boolean;
                        mandatory true;
                        description "*";
                    }
                    leaf tz-posix {
                        type string;
                        mandatory true;
                        description "*";
                    }
                    leaf tz-database {
                        type string;
                        mandatory true;
                        description "*";
                    }
                }
                container ntp-server-option {
                    description "*";
                    leaf enable {
                        type boolean;
                        mandatory true;
                        description "*";
                    }
                    list ntp-server {
                        key ntp-serv-id;
                        description "*";
                        leaf ntp-serv-id {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf ntp-serv-addr {
                            type inet:ipv6-address;
                            mandatory true;
                            description "*";
                        }
                        leaf ntp-serv-mul-addr {
                            type inet:ipv6-address;
                            mandatory true;
                            description "*";
                        }
                        leaf ntp-serv-fqdn {
                            type string;
                            mandatory true;
                            description "*";
                        }
                    }
                }
```

```
                container sntp-server-option {
                    description "*";
                    leaf enable {
                        type boolean;
                        mandatory true;
                        description "*";
                    }
                    list sntp-server {
                        key sntp-serv-id;
                        description "*";
                        leaf sntp-serv-id {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf sntp-serv-addr {
                            type inet:ipv6-address;
                            mandatory true;
                            description "*";
                        }
                    }
                }
                container network-boot-option {
                    description "*";
                    leaf enable {
                        type boolean;
                        mandatory true;
                        description "*";
                    }
                    list boot-file {
                        key boot-file-id;
                        description "*";
                        leaf boot-file-id {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf-list suitable-arch-type {
                            type uint16;
                            description "*";
                        }
                        leaf-list suitable-net-if {
                            type uint32;
                            description "*";
                        }
                        leaf boot-file-url {
                            type string;
                            mandatory true;
```

```
                        description "*";
                    }
                    list boot-file-paras {
                        key para-id;
                        description "*";
                        leaf para-id {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf parameter {
                            type string;
                            mandatory true;
                            description "*";
                        }
                    }
                }
            }
            container dslite-option {
                description "*";
                leaf enable {
                    type boolean;
                    mandatory true;
                    description "*";
                }
                leaf dslite-aftr-name {
                    type string;
                    mandatory true;
                    description "*";
                }
            }
            container kerberos-option {
                description "*";
                leaf enable {
                    type boolean;
                    mandatory true;
                    description "*";
                }
                leaf default-realm-name {
                    type string;
                    mandatory true;
                    description "*";
                }
                list kdc-info {
                    key kdc-id;
                    description "*";
                    leaf kdc-id {
                        type uint8;
```

```
                            mandatory true;
                            description "*";
                        }
                        leaf priority {
                            type uint16;
                            mandatory true;
                            description "*";
                        }
                        leaf weight {
                            type uint16;
                            mandatory true;
                            description "*";
                        }
                        leaf transport-type {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf port-number {
                            type uint16;
                            mandatory true;
                            description "*";
                        }
                        leaf kdc-ipv6-addr {
                            type inet:ipv6-address;
                            mandatory true;
                            description "*";
                        }
                        leaf realm-name {
                            type string;
                            mandatory true;
                            description "*";
                        }
                    }
                }
                container addr-selection-option {
                    description "*";
                    leaf enable {
                        type boolean;
                        mandatory true;
                        description "*";
                    }
                    leaf a-bit-set {
                        type boolean;
                        mandatory true;
                        description "*";
                    }
                    leaf p-bit-set {
```

```
                        type boolean;
                        mandatory true;
                        description "*";
                    }
                    list policy-table {
                        key policy-id;
                        description "*";
                        leaf policy-id {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf label {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf precedence {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf prefix-len {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf prefix {
                            type inet:ipv6-prefix;
                            mandatory true;
                            description "*";
                        }
                    }
                }
                container sol-max-rt-option {
                    description "*";
                    leaf enable {
                        type boolean;
                        mandatory true;
                        description "*";
                    }
                    leaf sol-max-rt-value {
                        type yang:timeticks;
                        mandatory true;
                        description "*";
                    }
                }
                container inf-max-rt-option {
```

```
                    description "*";
                    leaf enable {
                        type boolean;
                        mandatory true;
                        description "*";
                    }
                    leaf inf-max-rt-value {
                        type yang:timeticks;
                        mandatory true;
                        description "*";
                    }
                }
                container pcp-server-option {
                    description "*";
                    leaf enable {
                        type boolean;
                        mandatory true;
                        description "*";
                    }
                    list pcp-server {
                        key pcp-serv-id;
                        description "*";
                        leaf pcp-serv-id {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf pcp-serv-addr {
                            type inet:ipv6-address;
                            mandatory true;
                            description "*";
                        }
                    }
                }
                container s46-rule-option {
                    description "*";
                    leaf enable {
                        type boolean;
                        mandatory true;
                        description "*";
                    }
                    list s46-rule {
                        key rule-id;
                        description "*";
                        leaf rule-id {
                            type uint8;
                            mandatory true;
                            description "*";
```

```
                        }
                        leaf rule-type {
                            type enumeration {
                            enum "BMR";
                            enum "FMR";
                            }
                            mandatory true;
                            description "*";
                        }
                        leaf prefix4-len {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf ipv4-prefix {
                            type inet:ipv4-prefix;
                            mandatory true;
                            description "*";
                        }
                        leaf prefix6-len {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf ipv6-prefix {
                            type inet:ipv6-prefix;
                            mandatory true;
                            description "*";
                        }
                        uses portset-para;
                    }
                }
                container s46-br-option {
                    description "*";
                    leaf enable {
                        type boolean;
                        mandatory true;
                        description "*";
                    }
                    list br {
                        key br-id;
                        description "*";
                        leaf br-id {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf br-ipv6-addr {
```

```
                    type inet:ipv6-address;
                    mandatory true;
                    description "*";
                }
            }
        }
        container s46-dmr-option {
            leaf enable {
                type boolean;
                mandatory true;
                description "*";
            }
            list dmr {
                key dmr-id;
                description "*";
                leaf dmr-id {
                    type uint8;
                    mandatory true;
                    description "*";
                }
                leaf dmr-prefix-len {
                    type uint8;
                    mandatory true;
                    description "*";
                }
                leaf dmr-ipv6-prefix {
                    type inet:ipv6-prefix;
                    mandatory true;
                    description "*";
                }
            }
        }
        container s46-v4-v6-binding-option {
            description "*";
            leaf enable {
                type boolean;
                mandatory true;
                description "*";
            }
            list ce {
                key ce-id;
                description "*";
                leaf ce-id {
                    type uint8;
                    mandatory true;
                    description "*";
                }
                leaf ipv4-addr {
```

```
                            type inet:ipv4-address;
                            mandatory true;
                            description "*";
                        }
                        leaf bind-prefix6-len {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf bind-ipv6-prefix {
                            type inet:ipv6-prefix;
                            mandatory true;
                            description "*";
                        }
                        uses portset-para;
                    }
                }
            }
        }
            container network-range {
                description "*";
                leaf-list option-set {
                    type uint8;
                    description "*";
                }
                list network-range {
                    key network-range-id;
                    description "*";
                    leaf network-range-id {
                        type uint8;
                        mandatory true;
                        description "*";
                    }
                    leaf network-description {
                        type string;
                        mandatory true;
                        description "*";
                    }
                    leaf network-prefix {
                        type inet:ipv6-prefix;
                        mandatory true;
                        description "*";
                    }
                    leaf inherit-option-set {
                        type boolean;
                        mandatory true;
                        description "*";
                    }
```

```
                    leaf-list option-set {
                        type uint8;
                        description "*";
                    }
                    container address-pools {
                        description "A container describes
                        the DHCPv6 server's address pools.";
                        list address-pool {
                            key pool-id;
                            description "A DHCPv6 server can
                            be configured with several address
                            pools. This list defines such
                            address pools which are distinguish
                            by the key called 'pool-name'.";
                            leaf pool-id {
                                type uint8;
                                mandatory true;
                                description "*";
                            }
                            leaf pool-prefix {
                                type inet:ipv6-prefix;
                                mandatory true;
                                description "*";
                            }
                            leaf start-address {
                                type inet:ipv6-address-no-zone;
                                mandatory true;
                                description "*";
                            }
                            leaf end-address {
                                type inet:ipv6-address-no-zone;
                                mandatory true;
                                description "*";
                            }
                            leaf preferred-lifetime {
                                type yang:timeticks;
                                mandatory true;
                                description "*";
                            }
                            leaf valid-lifetime {
                                type yang:timeticks;
                                mandatory true;
                                description "*";
                            }
                            leaf total-ipv6-count {
                                type uint64;
                                config "false";
                                mandatory true;
```

```
                                    description "*";
                                }
                                leaf used-ipv6-count {
                                    type uint64;
                                    config "false";
                                    mandatory true;
                                    description "*";
                                }
                                leaf utilization-ratio {
                                    type threshold;
                                    mandatory true;
                                    description "*";
                                }
                                leaf inherit-option-set {
                                    type boolean;
                                    mandatory true;
                                    description "*";
                                }
                                leaf-list option-set {
                                    type uint8;
                                    description "*";
                                }
                                container reserved-addresses {
                                    description "*";
                                    list static-binding {
                                        key cli-id;
                                        description "*";
                                        leaf cli-id {
                                            type uint32;
                                            mandatory true;
                                            description "*";
                                        }
                                        leaf duid {
                                            type duidtype;
                                            mandatory true;
                                            description "DHCP Unique
                                            Identifer";
                                        }
                                        leaf-list reserv-addr {
                                            type inet:ipv6-address;
                                            description "*";
                                        }
                                    }
                                    leaf-list other-reserv-addr {
                                        type inet:ipv6-address;
                                        description "*";
                                    }
                                }
```

```
                        }
                        list binding-info {
                            key cli-id;
                            config "false";
                            description "A list records a binding
                            information for each DHCPv6 client
                            that has already been allocated IPv6
                            addresses.";
                            leaf cli-id {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf duid {
                                type duidtype;
                                mandatory true;
                                                                        description
"DHCP Unique Identifer";
                                                                    }
                            list cli-ia {
                                key iaid;
                                description "*";
                                leaf ia-type {
                                    type string;
                                    mandatory true;
                                    description "*";
                                }
                                leaf iaid {
                                    type uint32;
                                    mandatory true;
                                    description "*";
                                }
                                leaf-list cli-addr {
                                    type inet:ipv6-address;
                                    description "*";
                                }
                                leaf pool-id {
                                    type uint8;
                                    mandatory true;
                                    description "*";
                                }
                            }
                        }
                    }
                    container prefix-pools {
                        description "If a server supports prefix
                        delegation function, this container will
                        be used to define the delegating router's
```

```
                          refix pools.";
```

```
                        list prefix-pool {
                            key pool-id;
                            description "Similar to server's
                            address pools, a delegating router
                            can also be configured with multiple
                            prefix pools specified by a list
                            called 'prefix-pool'.";
                            leaf pool-id {
                                type uint8;
                                mandatory true;
                                description "*";
                            }
                            leaf prefix {
                                type inet:ipv6-prefix;
                                mandatory true;
                                description "*";
                            }
                            leaf prefix-length {
                                type uint8;
                                mandatory true;
                                description "*";
                            }
                            leaf preferred-lifetime {
                                type yang:timeticks;
                                mandatory true;
                                description "*";
                            }
                            leaf valid-lifetime {
                                type yang:timeticks;
                                mandatory true;
                                description "*";
                            }
                            leaf utilization-ratio {
                                type threshold;
                                mandatory true;
                                description "*";
                            }
                            leaf inherit-option-set {
                                type boolean;
                                mandatory true;
                                description "*";
                            }
                            leaf-list option-set {
                                type uint8;
                                description "*";
                            }
                            container reserved-prefixes {
                                description "*";
```

```
                                   list static-binding {
                                       key cli-id;
                                       description "*";
                                       leaf cli-id {
                                           type uint32;
                                           mandatory true;
                                           description "*";
                                       }
                                       leaf duid {
                                           type duidtype;
                                           mandatory true;
                                           description "DHCP Unique

Identifer";
                                       }
                                       leaf reserv-prefix-len {
                                           type uint8;
                                           mandatory true;
                                           description "*";
                                       }
                                       leaf reserv-prefix {
                                           type inet:ipv6-prefix;
                                           mandatory true;
                                           description "*";
                                       }
                                   }
                                   leaf exclude-prefix-len {
                                       type uint8;
                                       mandatory true;
                                       description "*";
                                   }
                                   leaf exclude-prefix {
                                       type inet:ipv6-prefix;
                                       mandatory true;
                                       description "*";
                                   }
                                   list other-reserv-prefix {
                                       key reserv-id;
                                       description "*";
                                       leaf reserv-id {
                                           type uint8;
                                           mandatory true;
                                           description "*";
                                       }
                                       leaf prefix-len {
                                           type uint8;
                                           mandatory true;
                                           description "*";
```

```
                                    }
```

```
                    leaf prefix {
                        type inet:ipv6-prefix;
                        mandatory true;
                        description "*";
                    }
                }
            }
        }
        list binding-info {
            key cli-id;
            config "false";
            description "A list records a
            binding information for each
            DHCPv6 client that has already
            been allocated IPv6 addresses.";
            leaf cli-id {
                type uint32;
                mandatory true;
                description "*";
            }
            leaf duid {
                type duidtype;
                manadatory true;
                description "DHCP Unique
                Identifier";
            }
            list cli-iapd {
                key iaid;
                description "*";
                leaf iaid {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf-list cli-prefix {
                    type inet:ipv6-prefix;
                    description "*";
                }
                leaf-list cli-prefix-len {
                    type uint8;
                    description "*";
                }
                leaf pool-id {
                    type uint8;
                    mandatory true;
                    description "*";
                }
            }
```

```
                        }
                    }
                    container hosts {
                        description "*";
                        list host {
                            key cli-id;
                            description "*";
                            leaf cli-id {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf duid {
                                type duidtype;
                                mandatory true;
                                description "DHCP Unique
                                Identifer";
                            }
                            leaf inherit-option-set {
                                type boolean;
                                mandatory true;
                                description "*";
                            }
                            leaf-list option-set {
                                type uint8;
                                description "*";
                            }
                            leaf nis-domain-name {
                                type string;
                                description "*";
                            }
                            leaf nis-plus-domain-name {
                                type string;
                                description "*";
                            }
                        }
                    }
                }
            }
            container relay-opaque-paras {
                description "This container contains some
                opaque values in Relay Agent options that
                need to be configured on the server side
                only for value match. Such Relay Agent
                options include Interface-Id option,
                Remote-Id option and Subscriber-Id option.";
                list relays {
                    key relay-name;
```

```
                    description "*";
                    leaf relay-name {
                        type string;
                        mandatory true;
                        description "*";
                    }
                    list interface-info {
                        key if-name;
                        description "*";
                        leaf if-name {
                            type string;
                            mandatory true;
                            description "*";
                        }
                        leaf interface-id {
                            type string;
                            mandatory true;
                            description "*";
                        }
                    }
                    list subscribers {
                        key subscriber;
                        description "*";
                        leaf subscriber {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf subscriber-id {
                            type string;
                            mandatory true;
                            description "*";
                        }
                    }
                    list remote-host {
                        key ent-num;
                        description "*";
                        leaf ent-num {
                            type uint32;
                            mandatory true;
                            description "*";
                        }
                        leaf remote-id {
                            type string;
                            mandatory true;
                            description "*";
                        }
                    }
```

```
                }
            }
            container rsoo-enabled-options {
                description "*";
                list rsoo-enabled-option{
                    key option-code;
                    description "*";
                    leaf option-code {
                        type uint16;
                        mandatory true;
                        description "*";
                    }
                    leaf description {
                        type string;
                        mandatory true;
                        description "*";
                    }
                }
            }
            container packet-stats {
                config "false";
                description "A container presents
                the packet statistics related to
                the DHCPv6 server.";
                leaf solicit-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf request-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf renew-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf rebind-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf decline-count {
                    type uint32;
                    mandatory true;
                    description "*";
```

```
                }
                leaf release-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf info-req-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf advertise-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf confirm-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf reply-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf reconfigure-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf relay-forward-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf relay-reply-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
            }
        }
    }
    container relay {
        if-feature dhcpv6-relay;
        description "relay portion";
        container relay-attributes {
```

```
            description "A container describes
            some basic attributes of the relay
            agent including some relay agent
            specific options data that need to
            be configured previously. Such options
            include Remote-Id option and
            Subscriber-Id option.";
            leaf name {
                type string;
                mandatory true;
                description "*";
            }
            leaf enable {
                type boolean;
                mandatory true;
                description "*";
            }
            leaf ipv6-address {
                type inet:ipv6-address;
                mandatory true;
                description "*";
            }
            leaf description {
                type string;
                description "*";
            }
            leaf-list dest-addrs {
                type inet:ipv6-address;
                description "Each DHCPv6 relay agent may
                be configured with a list of destination
                addresses. This node defines such a list
                of IPv6 addresses that may include
                unicast addresses, multicast addresses
                or other addresses.";
            }
            list subscribers {
                key subscriber;
                description "*";
                leaf subscriber {
                    type uint8;
                    mandatory true;
                    description "*";
                }
                leaf subscriber-id {
                    type string;
                    mandatory true;
                    description "*";
                }
```

```
            }
            list remote-host {
                key ent-num;
                description "*";
                leaf ent-num {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf remote-id {
                    type string;
                    mandatory true;
                    description "*";
                }
            }
            uses vendor-infor;
            container relay-supplied-options-option {
                description "*";
                list rsoo-set {
                    key rsoo-set-id;
                    description "*";
                    leaf rsoo-set-id {
                        type uint8;
                        mandatory true;
                        description "*";
                    }
                    container erp-local-domain-name-option {
                        description "*";
                        leaf enable {
                            type boolean;
                            mandatory true;
                            description "*";
                        }
                        list erp-for-client {
                            key cli-id;
                            description "*";
                            leaf cli-id {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf duid {
                                type duidtype;
                                mandatory true;
                                description "DHCP Unique
                                Identifer";
                            }
                            leaf erp-name {
```

```
                                type string;
                                mandatory true;
                                description "*";
                            }
                        }
                    }
                }
            }
            container relay-interfaces {
                description "It is a container that
                defines common configuration and
                state parameters in the interfaces
                of a DHCPv6 relay agent. In this YANG
                data model for DHCPv6 relay agent,
                the parameters are configured in a
                per-interface manner.";
                list relay-if {
                    key if-name;
                    description "A list describes a
                    specific interface and its
                    corresponding parameters. Here
                    we use a string called 'ifName'
                    as the key of list.";
                    leaf if-name {
                        type string;
                        mandatory true;
                        description "*";
                    }
                    leaf enable {
                        type boolean;
                        mandatory true;
                        description "*";
                    }
                    leaf interface-id {
                        type string;
                        description "*";
                    }
                    leaf-list rsoo-set {
                        type uint8;
                        description "*";
                    }
                    list pd-route {
                        key pd-route-id;
                        description "*";
                        leaf pd-route-id {
                            type uint8;
                            mandatory true;
                            description "*";
```

```
                        }
                        leaf requesting-router- d {
                            type uint32;
                            mandatory true;
                            description "*";
                        }
                        leaf delegating-router-id {
                            type uint32;
                            mandatory true;
                            description "*";
                        }
                        leaf next-router {
                            type inet:ipv6-address;
                            mandatory true;
                            description "*";
                        }
                        leaf last-router {
                            type inet:ipv6-address;
                            mandatory true;
                            description "*";
                        }
                    }
                    list next-entity {
                        key dest-addr;
                        description "This node defines
                        a list that is used to describe
                        the next hop entity of this
                        relay distinguished by their
                        addresses.";
                        leaf dest-addr {
                            type inet:ipv6-address;
                            mandatory true;
                            description "*";
                        }
                        leaf available {
                            type boolean;
                            mandatory true;
                            description "*";
                        }
                        leaf multicast {
                            type boolean;
                            mandatory true;
                            description "*";
                        }
                        leaf server {
                            type boolean;
                            mandatory true;
                            description "*";
```

```
                        }
                        container packet-stats {
                            config "false";
                            description "*";
                            leaf cli-packet-rvd-count {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf solicit-rvd-count {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf request-rvd-count {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf renew-rvd-count {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf rebind-rvd-count {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf decline-rvd-count {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf release-rvd-count {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf info-req-rvd-count {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf relay-for-rvd-count {
                                type uint32;
                                mandatory true;
                                description "*";
```

```
                            }
                            leaf relay-rep-rvd-count {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf packet-to-cli-count {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf adver-sent-count {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf confirm-sent-count {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf reply-sent-count {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf reconfig-sent-count {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf relay-for-sent-count {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                            leaf relay-rep-sent-count {
                                type uint32;
                                mandatory true;
                                description "*";
                            }
                        }
                    }
                }
            }
            container relay-stats {
                config "false";
                description "*";
```

```
                leaf cli-packet-rvd-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf relay-for-rvd-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf relay-rep-rvd-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf packet-to-cli-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf relay-for-sent-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf relay-rep-sent-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf discarded-packet-count {
                    type uint32;
                    mandatory true;
                    description "*";
                }
            }
        }
    }
    container client {
        if-feature dhcpv6-client;
        description "DHCPv6 Client Portion";
        container client-interfaces {
            description "A client may have several
            interfaces, it is more reasonable to
            configure and manage parameters on
            the interface-level. This container
            includes configuration and state data
            of a DHCPv6 client in a per-interface
```

```
            manner.";
            list client-if {
                key if-name;
                description "The list defines a
                specific client interface and its
                data. Different interfaces are
                distinguished by the key which
                is a configurable string value.";
                leaf if-name {
                    type string;
                    mandatory true;
                    description "interface name";
                }
                leaf cli-id {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf duid {
                    type duidtype;
                    mandatory true;
                    description "DHCP Unique
                    Identifer";
                }
                leaf enable {
                    type boolean;
                    mandatory true;
                    description "*";
                }
                leaf description {
                    type string;
                    description "*";
                }
                leaf pd-function {
                    type boolean;
                    description "Whether the client
                    can act as a requesting router
                    to request prefixes using prefix
                    delegation ([RFC3633]).";
                    mandatory true;
                }
                leaf rapid-commit {
                    type boolean;
                    description "'1' indicates a client
                    can initiate a Solicit-Reply message
                    exchange by adding a Rapid Commit
                    option in Solicit message. '0' means
                    the client is not allowed to add a
```

```
                    Rapid Commit option to request
                    addresses in a two-message
                    exchange pattern.";
                    mandatory true;
                }
                container mo-tab {
                    description "The management tab
                    label indicates the operation
                    mode of the DHCPv6 client. 'm'=1
                    and 'o'=1 indicate the client
                    will use DHCPv6 to obtain all
                    the configuration data. 'm'=1
                    and 'o'=0 are a meaningless
                    combination. 'm'=0 and 'o'=1
                    indicate the client will use
                    stateless DHCPv6 to obtain
                    configuration data apart from
                    addresses/prefixes data.
                    'm'=0 and 'o'=0 represent the
                    client will not use DHCPv6
                    but use SLAAC to achieve
                    configuration.";
                    leaf m-tab {
                        type boolean;
                        mandatory true;
                        description "*";
                    }
                    leaf o-tab {
                        type boolean;
                        mandatory true;
                        description "*";
                    }
                }
                container oro-options {
                    description "*";
                    list oro-option {
                        key option-code;
                        description "*";
                        leaf option-code {
                            type uint16;
                            mandatory true;
                            description "*";
                        }
                        leaf description {
                            type string;
                            mandatory true;
                            description "*";
                        }
```

```
                    }
                }
                container client-configured-options {
                    description "*";
                    list new-cli-option {
                        key option-code;
                        description "*";
                        leaf option-code {
                            type uint16;
                            manadatory true;
                            description "*";
                        }
                        leaf option-name {
                            type string;
                            manadatory true;
                            description "*";
                        }
                        leaf option-description {
                            type string;
                            manadatory true;
                            description "*";
                        }
                        leaf option-reference {
                            type string;
                            description "*";
                        }
                        leaf option-value {
                            type string;
                            manadatory true;
                            description "*";
                        }
                    }
                    container user-class-option {
                        description "*";
                        leaf enable {
                            type boolean;
                            mandatory true;
                            description "*";
                        }
                        list user-class {
                            key user-class-id;
                            description "*";
                            leaf user-class-id {
                                type uint8;
                                mandatory true;
                                description "*";
                            }
                            leaf user-class-info {
```

```
                    type string;
                    mandatory true;
                    description "*";
                }
            }
        }
        container vendor-class-option {
            description "*";
            leaf enable {
                type boolean;
                mandatory true;
                description "*";
            }
            leaf ent-num {
                type uint32;
                mandatory true;
                description "*";
            }
            leaf-list data {
                type string;
                description "*";
            }
        }
        container client-fqdn-option {
            description "*";
            leaf enable {
                type boolean;
                mandatory true;
                description "*";
            }
            leaf fqdn {
                type string;
                mandatory true;
                description "*";
            }
            leaf server-initiate-update {
                type boolean;
                mandatory true;
                description "*";
            }
            leaf client-initiate-update {
                type boolean;
                mandatory true;
                description "*";
            }
        }
        container client-architecture-type-option {
            description "*";
```

```
                        leaf enable {
                            type boolean;
                            mandatory true;
                            description "*";
                        }
                        list architecture-types {
                            key type-id;
                            description "*";
                            leaf type-id {
                                type uint16;
                                mandatory true;
                                description "*";
                            }
                            leaf most-preferred {
                                type boolean;
                                mandatory true;
                                description "*";
                            }
                        }
                    }
                    container client-network-interface-option {
                        description "*";
                        leaf enable {
                            type boolean;
                            mandatory true;
                            description "*";
                        }
                        leaf type {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf major {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf minor {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                    }
                    container kerberos-principal-name-option {
                        description "*";
                        leaf enable {
                            type boolean;
                            mandatory true;
```

```
                            description "*";
                        }
                        leaf principal-name {
                            type string;
                            mandatory true;
                            description "*";
                        }
                    }
                    container client-link-layer-addr-option {
                        description "*";
                        leaf enable {
                            type boolean;
                            mandatory true;
                            description "*";
                        }
                        leaf link-layer-type {
                            type uint16;
                            mandatory true;
                            description "*";
                        }
                        leaf link-layer-addr {
                            type string;
                            mandatory true;
                            description "*";
                        }
                    }
                }
                container identity-associations {
                    config "false";
                    description "IA is a construct through
                    which a server and a client can identify,
                    group, and manage a set of related IPv6
                    addresses. The key of the list is a
                    4-byte number IAID defined in [RFC3315].";
                    list identity-association {
                        key iaid;
                        description "*";
                        leaf iaid {
                            type uint32;
                            mandatory true;
                            description "*";
                        }
                        leaf ia-type {
                            type string;
                            mandatory true;
                            description "*";
                        }
                        leaf-list ipv6-addr {
```

```
                        type inet:ipv6-address;
                        description "*";
                    }
                    leaf-list ipv6-prefix {
                        type inet:ipv6-prefix;
                        description "*";
                    }
                    leaf-list prefix-length {
                        type uint8;
                        description "*";
                    }
                    leaf t1-time {
                        type yang:date-and-time;
                        mandatory true;
                        description "*";
                    }
                    leaf t2-time {
                        type yang:date-and-time;
                        mandatory true;
                        description "*";
                    }
                    leaf preferred-lifetime {
                        type yang:timeticks;
                        mandatory true;
                        description "*";
                    }
                    leaf valid-lifetime {
                        type yang:timeticks;
                        mandatory true;
                        description "*";
                    }
                }
            }
            container if-other-paras {
                config "false";
                description "A client can obtain
                extra configuration data other than
                address and prefix information through
                DHCPv6. This container describes such
                data the client was configured. The
                potential configuration data may
                include DNS server addresses, SIP
                server domain names, etc.";
                leaf-list uni-dhcpv6-serv-addr {
                    type inet:ipv6-address;
                    description "*";
                }
                container dns-paras {
```

```
                    description "*";
                    leaf domain-search-list {
                        type string;
                        mandatory true;
                        description "*";
                    }
                    list dns-servers {
                        key dns-serv-id;
                        description "*";
                        leaf dns-serv-id {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf dns-serv-addr {
                            type inet:ipv6-address;
                            mandatory true;
                            description "*";
                        }
                    }
                }
                container sip-paras {
                    description "*";
                    list sip-servers {
                        key sip-serv-id;
                        description "*";
                        leaf sip-serv-id {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf sip-serv-addr {
                            type inet:ipv6-address;
                            mandatory true;
                            description "*";
                        }
                        leaf sip-serv-domain-name {
                            type string;
                            mandatory true;
                            description "*";
                        }
                    }
                }
                container nis-paras {
                    description "*";
                    leaf nis-domain-name {
                        type string;
                        mandatory true;
```

```
                    description "*";
                }
                list nis-server {
                    key nis-serv-id;
                    description "*";
                    leaf nis-serv-id {
                        type uint8;
                        mandatory true;
                        description "*";
                    }
                    leaf nis-serv-addr {
                        type inet:ipv6-address;
                        mandatory true;
                        description "*";
                    }
                }
            }
            container nis-plus-paras {
                description "*";
                leaf nis-plus-domain-name {
                    type string;
                    mandatory true;
                    description "*";
                }
                list nis-plus-server {
                    key nis-plus-serv-id;
                    description "*";
                    leaf nis-plus-serv-id {
                        type uint8;
                        mandatory true;
                        description "*";
                    }
                    leaf nis-plus-serv-addr {
                        type inet:ipv6-address;
                        mandatory true;
                        description "*";
                    }
                }
            }
            leaf info-refresh-time {
                type yang:timeticks;
                description "*";
            }
            container time-zone-paras {
                description "*";
                leaf tz-posix {
                    type string;
                    mandatory true;
```

```
                        description "*";
                    }
                    leaf tz-database {
                        type string;
                        mandatory true;
                        description "*";
                    }
                }
                leaf cli-fqdn {
                    type string;
                    description "*";
                }
                container ntp-paras {
                    description "*";
                    list ntp-server {
                        key ntp-serv-id;
                        description "*";
                        leaf ntp-serv-id {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf ntp-serv-addr {
                            type inet:ipv6-address;
                            mandatory true;
                            description "*";
                        }
                        leaf ntp-serv-mul-addr {
                            type inet:ipv6-address;
                            mandatory true;
                            description "*";
                        }
                        leaf ntp-serv-fqdn {
                            type string;
                            mandatory true;
                            description "*";
                        }
                    }
                }
                container sntp-paras {
                    description "*";
                    list sntp-server {
                        key sntp-serv-id;
                        description "*";
                        leaf sntp-serv-id {
                            type uint8;
                            mandatory true;
                            description "*";
```

```
                }
                leaf sntp-serv-addr {
                    type inet:ipv6-address;
                    mandatory true;
                    description "*";
                }
            }
        }
        container network-boot-paras {
            description "*";
            list boot-file {
                key boot-file-id;
                description "*";
                leaf boot-file-id {
                    type uint8;
                    mandatory true;
                    description "*";
                }
                leaf-list suitable-arch-type {
                    type uint16;
                    description "*";
                }
                leaf-list suitable-net-if {
                    type uint32;
                    description "*";
                }
                leaf boot-file-url {
                    type string;
                    mandatory true;
                    description "*";
                }
                list boot-file-paras {
                    key para-id;
                    description "*";
                    leaf para-id {
                        type uint8;
                        mandatory true;
                        description "*";
                    }
                    leaf parameter {
                        type string;
                        mandatory true;
                        description "*";
                    }
                }
            }
        }
        container kerberos-paras {
```

```
                        description "*";
                        leaf default-realm-name {
                            type string;
                            mandatory true
                            description "*";
                        }
                        list kdc-info {
                            key kdc-id;
                            description "*";
                            leaf kdc-id {
                                type uint8;
                                mandatory true;
                                description "*";
                            }
                            leaf priority {
                                type uint16;
                                mandatory true;
                                description "*";
                            }
                            leaf weight {
                                type uint16;
                                mandatory true;
                                description "*";
                            }
                            leaf transport-type {
                                type uint8;
                                mandatory true;
                                description "*";
                            }
                            leaf port-number {
                                type uint16;
                                mandatory true;
                                description "*";
                            }
                            leaf kdc-ipv6-addr {
                                type inet:ipv6-address;
                                mandatory true;
                                description "*";
                            }
                            leaf realm-name {
                                type string;
                                mandatory true;
                                description "*";
                            }
                        }
                    }
                    container addr-selection-paras {
                        description "*";
```

```
                    leaf automatic-row-add {
                        type boolean;
                        mandatory true;
                        description "*";
                    }
                    leaf prefer-temporary-addr {
                        type boolean;
                        mandatory true;
                        description "*";
                    }
                    list policy-table {
                        key policy-id;
                        description "*";
                        leaf policy-id {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf label {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf precedence {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf prefix-len {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        leaf prefix {
                            type inet:ipv6-prefix;
                            mandatory true;
                            description "*";
                        }
                    }
                }
                leaf sol-max-rt {
                    type yang:timeticks;
                    mandatory true;
                    description "*";
                }
                leaf inf-max-rt {
                    type yang:timeticks;
                    mandatory true;
```

```
                        description "*";
                    }
                    container pcp-server-paras {
                        description "*";
                        list pcp-server {
                            key pcp-serv-id;
                            description "*";
                            leaf pcp-serv-id {
                                type uint8;
                                mandatory true;
                                description "*";
                            }
                            leaf pcp-serv-addr {
                                type inet:ipv6-address;
                                mandatory true;
                                description "*";
                            }
                        }
                    }
                    container s46-rule-paras {
                        description "*";
                        list s46-rule {
                            key rule-id;
                            description "*";
                            leaf rule-id {
                                type uint8;
                                mandatory true;
                                description "*";
                            }
                            leaf rule-type {
                                type enumeration {
                                enum "BMR";
                                enum "FMR";
                                }
                                mandatory true;
                                description "*";
                            }
                            leaf ea-len {
                                type uint8;
                                mandatory true;
                                description "*";
                            }
                            leaf prefix4-len {
                                type uint8;
                                mandatory true;
                                description "*";
                            }
                            leaf ipv4-prefix {
```

```
                    type inet:ipv4-prefix;
                    mandatory true;
                    description "*";
                }
                leaf prefix6-len {
                    type uint8;
                    mandatory true;
                    description "*";
                }
                leaf ipv6-prefix {
                    type inet:ipv6-prefix;
                    mandatory true;
                    description "*";
                }
                uses portset-para;
            }
        }
        container s46-br-paras {
            description "*";
            list br {
                key br-id;
                description "*";
                leaf br-id {
                    type uint8;
                    mandatory true;
                    description "*";
                }
                leaf br-ipv6-addr {
                    type inet:ipv6-address;
                    mandatory true;
                    description "*";
                }
            }
        }
        container s46-dmr-paras {
            description "*";
            list dmr {
                key dmr-id;
                description "*";
                leaf dmr-id {
                    type uint8;
                    mandatory true;
                    description "*";
                }
                leaf dmr-prefix-len {
                    type uint8;
                    mandatory true;
                    description "*";
```

```
                            }
                            leaf dmr-ipv6-prefix {
                                type inet:ipv6-prefix;
                                mandatory true;
                                description "*";
                            }
                        }
                    }
                    container s46-v4-v6-binding-paras {
                        description "*";
                        leaf ipv4-addr {
                            type inet:ipv4-address;
                            mandatory true;
                            description "*";
                        }
                        leaf bind-prefix6-len {
                            type uint8;
                            mandatory true;
                            description "*";
                        }
                        uses portset-para;
                        leaf erp-local-domain-name {
                            type string;
                            mandatory true;
                            description "*";
                        }
                    }
                    container supported-options {
                        description "*";
                        list supported-option {
                            key option-code;
                            description "*";
                            leaf option-code {
                                type uint16;
                                mandatory true;
                                description "*";
                            }
                            leaf description {
                                type string;
                                mandatory true;
                                description "*";
                            }
                        }
                    }
                }
                container packet-stats {
                    config "false";
                    description "A container records
```

```
                    all the packet status information
                    of a specific interface.";
                    leaf solicit-count {
                        type uint32;
                        mandatory true;
                        description "*";
                    }
                    leaf request-count {
                        type uint32;
                        mandatory true;
                        description "*";
                    }
                    leaf renew-count {
                        type uint32;
                        mandatory true;
                        description "*";
                    }
                    leaf rebind-count {
                        type uint32;
                        mandatory true;
                        description "*";
                    }
                    leaf decline-count {
                        type uint32;
                        mandatory true;
                        description "*";
                    }
                    leaf release-count {
                        type uint32;
                        mandatory true;
                        description "*";
                    }
                    leaf info-req-count {
                        type uint32;
                        mandatory true;
                        description "*";
                    }
                    leaf advertise-count {
                        type uint32;
                        mandatory true;
                        description "*";
                    }
                    leaf confirm-count {
                        type uint32;
                        mandatory true;
                        description "*";
                    }
                    leaf reply-count {
```

```
                        type uint32;
                        mandatory true;
                        description "*";
                    }
                    leaf reconfigure-count {
                        type uint32;
                        mandatory true;
                        description "*";
                    }
                }
            }
        }
    }

/*
 * Notifications
 */

    notification notifications {
        container dhcpv6-server-event {
            if-feature dhcpv6-server;
            description "server portion";
            container pool-running-out {
                description "raised when the address/prefix pool is going to
                run out.  A threshold for utilization ratio of the pool has
been
                defined in the server feature so that it will notify the
                administrator when the utilization ratio reaches the threshold,
                and such threshold is a settable parameter";
                leaf utilization-ratio {
                    type uint16;
                    mandatory true;
                    description "*";
                }
                leaf duid {
                    type duidtype;
                    mandatory true;
                    description "DHCP Unique
                    Identifer";
                }
                leaf serv-name {
                    type string;
                    description "*";
                }
                leaf pool-name {
                    type string;
                    mandatory true;
                    description "*";
```

```
                }
```

```
            }
            container invalid-client-detected {
                description "raised when the server has found a client
                which can be regarded as a potential attacker.  Some
                description could also be included.";
                leaf duid {
                    type duidtype;
                    mandatory true;
                    description "DHCP Unique
                    Identifer";
                }
                leaf description {
                    type string;
                    description "*";
                }
            }
        }
        container dhcpv6-relay-event {
            if-feature dhcpv6-relay;
            description "relay portion";
            container topo-changed {
                description "raised when the topology
                of the relay agent is changed.";
                leaf relay-if-name {
                    type string;
                    mandatory true;
                    description "*";
                }
                leaf first-hop {
                    type boolean;
                    mandatory true;
                    description "*";
                }
                leaf last-entity-addr {
                    type inet:ipv6-address;
                    mandatory true;
                    description "*";
                }
            }
        }
        container dhcpv6-client-event {
            if-feature dhcpv6-client;
            description "client portion";
            container ia-lease-event {
                description "raised when the
                client was allocated a new IA from
                the server or it renew/rebind/release
                its current IA";
```

```
                leaf event-type {
                    type enumeration{
                    enum "allocation";
                    enum "rebind";
                    enum "renew";
                    enum "release";
                    }
                    mandatory true;
                    description "*";
                }
                leaf duid {
                    type duidtype;
                    mandatory true;
                    description "DHCP Unique
                    Identifer";
                }
                leaf iaid {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf serv-name {
                    type string;
                    description "*";
                }
                leaf description {
                    type string;
                    description "*";
                }
            }
            container invalid-ia-detected {
                description "raised when the identity
                association of the client can be proved
                to be invalid.  Possible condition includes
                duplicated address, illegal address, etc.";
                leaf duid {
                    type duidtype;
                    mandatory true;
                    description "DHCP Unique
                    Identifer";
                }
                leaf cli-duid {
                    type uint32;
                    mandatory true;
                    description "*";
                }
                leaf iaid {
                    type uint32;
```

```
                    mandatory true;
                    description "*";
                }
                leaf serv-name {
                    type string;
                    description "*";
                }
                leaf description {
                    type string;
                    description "*";
                }
            }
            container retransmission-failed {
                description "raised when the retransmission
                mechanism defined in [RFC3315] is failed.";
                uses duid-para;
                leaf description {
                    type enumeration {
                        enum "MRC failed";
                        enum "MRD failed";
                    }
                    mandatory true;
                    description "*";
                }
            }
            container failed-status-turn-up {
                description "raised when the client receives
                a message includes an unsuccessful Status Code
                option.";
                leaf duid {
                    type duidtype;
                    mandatory true;
                    description "DHCP Unique
                    Identifer";
                }
                leaf status-code {
                    type enumeration {
                        enum "1" {
                            description "UnspecFail";
                        }
                        enum "2" {
                            description "NoAddrAvail";
                        }
                        enum "3" {
                            description "NoBinding";
                        }
                        enum "4" {
                            description "NotOnLink";
```

```
                        }
                        enum "5" {
                            description "UseMulticast";
                        }
                    }
                    mandatory true;
                    description "*";
                }
            }
        }
    }
}
<CODE ENDS>
```

5. Security Considerations (TBD)

TBD

6. IANA Considerations (TBD)

TBD

7. Acknowledgements

The authors would like to thank Qi Sun, Lishan Li and Sladjana Zoric for their valuable contributions to this work.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <http://www.rfc-editor.org/info/rfc3315>.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, DOI 10.17487/RFC3633, December 2003, <http://www.rfc-editor.org/info/rfc3633>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6087] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", RFC 6087, DOI 10.17487/RFC6087, January 2011, <http://www.rfc-editor.org/info/rfc6087>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <http://www.rfc-editor.org/info/rfc6991>.

8.2. Informative References

[RFC3319] Schulzrinne, H. and B. Volz, "Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers", RFC 3319, DOI 10.17487/RFC3319, July 2003, <http://www.rfc-editor.org/info/rfc3319>.

[RFC3646] Droms, R., Ed., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, DOI 10.17487/RFC3646, December 2003, <http://www.rfc-editor.org/info/rfc3646>.

[RFC3898] Kalusivalingam, V., "Network Information Service (NIS) Configuration Options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3898, DOI 10.17487/RFC3898, October 2004, <http://www.rfc-editor.org/info/rfc3898>.

[RFC4075] Kalusivalingam, V., "Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6", RFC 4075, DOI 10.17487/RFC4075, May 2005, <http://www.rfc-editor.org/info/rfc4075>.

[RFC4242] Venaas, S., Chown, T., and B. Volz, "Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 4242, DOI 10.17487/RFC4242, November 2005, <http://www.rfc-editor.org/info/rfc4242>.

[RFC4704] Volz, B., "The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option", RFC 4704, DOI 10.17487/RFC4704, October 2006, <http://www.rfc-editor.org/info/rfc4704>.

[RFC4833] Lear, E. and P. Eggert, "Timezone Options for DHCP", RFC 4833, DOI 10.17487/RFC4833, April 2007, <http://www.rfc-editor.org/info/rfc4833>.

[RFC5908] Gayraud, R. and B. Lourdelet, "Network Time Protocol (NTP) Server Option for DHCPv6", RFC 5908, DOI 10.17487/RFC5908, June 2010, <http://www.rfc-editor.org/info/rfc5908>.

[RFC5970] Huth, T., Freimann, J., Zimmer, V., and D. Thaler, "DHCPv6 Options for Network Boot", RFC 5970, DOI 10.17487/RFC5970, September 2010, <http://www.rfc-editor.org/info/rfc5970>.

[RFC6334] Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite", RFC 6334, DOI 10.17487/RFC6334, August 2011, <http://www.rfc-editor.org/info/rfc6334>.

[RFC6422] Lemon, T. and Q. Wu, "Relay-Supplied DHCP Options", RFC 6422, DOI 10.17487/RFC6422, December 2011, <http://www.rfc-editor.org/info/rfc6422>.

[RFC6440] Zorn, G., Wu, Q., and Y. Wang, "The EAP Re-authentication Protocol (ERP) Local Domain Name DHCPv6 Option", RFC 6440, DOI 10.17487/RFC6440, December 2011, <http://www.rfc-editor.org/info/rfc6440>.

[RFC6784] Sakane, S. and M. Ishiyama, "Kerberos Options for DHCPv6", RFC 6784, DOI 10.17487/RFC6784, November 2012, <http://www.rfc-editor.org/info/rfc6784>.

[RFC6939] Halwasia, G., Bhandari, S., and W. Dec, "Client Link-Layer Address Option in DHCPv6", RFC 6939, DOI 10.17487/RFC6939, May 2013, <http://www.rfc-editor.org/info/rfc6939>.

[RFC7078] Matsumoto, A., Fujisaki, T., and T. Chown, "Distributing Address Selection Policy Using DHCPv6", RFC 7078, DOI 10.17487/RFC7078, January 2014, <http://www.rfc-editor.org/info/rfc7078>.

[RFC7083] Droms, R., "Modification to Default Values of SOL_MAX_RT and INF_MAX_RT", RFC 7083, DOI 10.17487/RFC7083, November 2013, <http://www.rfc-editor.org/info/rfc7083>.

[RFC7291] Boucadair, M., Penno, R., and D. Wing, "DHCP Options for the Port Control Protocol (PCP)", RFC 7291, DOI 10.17487/RFC7291, July 2014, <http://www.rfc-editor.org/info/rfc7291>.

[RFC7598] Mrugalski, T., Troan, O., Farrer, I., Perreault, S., Dec, W., Bao, C., Yeh, L., and X. Deng, "DHCPv6 Options for Configuration of Softwire Address and Port-Mapped Clients", RFC 7598, DOI 10.17487/RFC7598, July 2015, <http://www.rfc-editor.org/info/rfc7598>.

Authors' Addresses

Yong Cui
Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6260-3059
Email: yong@csnet1.cs.tsinghua.edu.cn

Hao Wang
Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5822
Email: wangh13@mails.tsinghua.edu.cn

Linhui Sun
Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5822
Email: lh.sunlinh@gmail.com

Ted Lemon
Nominum, Inc.
950 Charter St.
Redwood City, CA 94043
USA

Email: Ted.Lemon@nominum.com

Ian Farrer
Deutsche Telekom AG
CTO-ATI, Landgrabenweg 151
Bonn, NRW 53227
Germany

Email: ian.farrer@telekom.de