

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: September 14, 2017

Q. Sun  
C. Xie  
China Telecom  
M. Boucadair  
Orange  
T. Peng  
Huawei Technologies  
Y. Lee  
Comcast  
March 13, 2017

**A YANG Data Model for Address Pool Management**  
**draft-sun-casm-address-pool-management-yang-00**

Abstract

This document specifies a YANG data model for IP address pool management. It can be used to automatically allocate, update and delete address pools in different devices of an underlying network.

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 September 14, 2017.

Copyright Notice

Copyright (c) 2017 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

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">2.</a>	Terminology . . . . .	<a href="#">3</a>
<a href="#">3.</a>	AddressPoolManagement Data Model . . . . .	<a href="#">3</a>
<a href="#">4.</a>	AddressPoolManagement YANG Module . . . . .	<a href="#">4</a>
<a href="#">5.</a>	Security Considerations . . . . .	<a href="#">13</a>
<a href="#">6.</a>	IANA Considerations . . . . .	<a href="#">13</a>
<a href="#">7.</a>	Contributors . . . . .	<a href="#">13</a>
<a href="#">8.</a>	Acknowledgements . . . . .	<a href="#">14</a>
<a href="#">9.</a>	References . . . . .	<a href="#">14</a>
<a href="#">9.1.</a>	Normative References . . . . .	<a href="#">14</a>
<a href="#">9.2.</a>	Informative References . . . . .	<a href="#">14</a>
	Authors' Addresses . . . . .	<a href="#">14</a>

## [1.](#) Introduction

IP address pool management is one of the basic elements to configure in a network to offer connectivity services to connected devices. Concretely, pools can be provisioned to DHCP servers, IPv4 service continuity devices (e.g., DS-Lite AFTR, NAT64), Carrier Grade NAT (CGN), Broadband Network Gateway (BNG), etc. Automated means to rationalize the management of address resources and to make sure the underlying routing and forwarding capabilities are appropriately configured, are helpful for operators. This document specifies a YANG data model for that purpose.

A device can be provisioned with a pool of addresses for various reasons: service requesting hosts with addresses or prefixes (e.g., DHCP server, Delegating router), inject appropriate routing entries (e.g., PE, BNG) given that prefix assignments and routing actions must be correlated otherwise delivery of connectivity service will fail. This document does not elaborate the usage of pools provisioned to a network element.

It is worth mentioning that: (1) current practices rely on static configuration which is prone to errors, (2) the level of route aggregation cannot be driven by PE routers without any hint(s) from an entity that has the visibility on aggregation policies and the status of prefixes, etc., and (3) relying on proprietary means to trigger the injection of routing entries may lead to undesired behavior such as to increase the size of routing table and forwarding table due to injecting very specific routes.



Within this document, an address pool usually contains the address pool type, start-address, end-address, its corresponding lifetime and the identification of the usage. Each address pool is represented by an Address Pool Entry (APE).

Pools may be specific to a service offered by a network or be valid for all services.

Pools can be added and/or withdrawn.

## 2. Terminology

The terminology for describing YANG data models is defined in [\[RFC6020\]](#).

The meaning of the symbols in the tree diagrams is as follows:

Brackets "[" and "]" enclose list keys.

Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.

Abbreviations before data node names: "rw" means configuration (read-write), "ro" state data (read-only).

Symbols after data node names: "?" means an optional node, "!" a container with presence, and "\*" denotes a "list" or "leaf-list".

Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").

Ellipsis ("...") stands for contents of subtrees that are not shown.

## 3. AddressPoolManagement Data Model

Two YANG modules are defined (Figure 1). The first module, "ietf-address-pool", defines generic address pool aspects which is common to all use cases. The second module, "ietf-address-pool-status", defines the status of the address pool.

```
module: ietf-address-pool
  +--rw address-pools
  |   +--rw address-pool* [address-pool-name]
  |       +--rw address-pool-name          string
  |       +--rw address-pool-id            uint32  default "0"
  |       +--rw device-id?                 string
  |       +--rw domain-name                string
```



```

|   +--rw address-pool-service* [service-name]
|   |   +--rw service-name      string
|   +--rw address-pool-entries
|       +--rw ipv4-address-range* [ipv4-address-range-name]
|           |   +--rw ipv4-address-range-name      string
|           |   +--rw ip-lower-address?            inet:ipv4-address-no-zone
|           |   +--rw ip-upper-address?            inet:ipv4-address-no-zone
|           |   +--rw usergateway?                  inet:ipv4-address-no-zone
|           |   +--rw gwnetmask?                    yang:dotted-quad
|           |   +--rw type?                          address-pool-type
|           |   +--rw lifetime?                      yang:date-and-time
|           |   +--rw instance?                      instance-type
|       +--rw warning-threshold-v4? percent
|       +--rw ipv6-prefix* [ipv6-prefix-name]
|           |   +--rw ipv6-prefix-name              string
|           |   +--rw ipv6-prefix?                  inet:ipv6-prefix
|           |   +--rw usergateway?                  inet:ipv6-address-no-zone
|           |   +--rw type?                          address-pool-type
|           |   +--rw lifetime?                      yang:date-and-time
|           |   +--rw instance?                      instance-type
|       +--rw warning-threshold-v6? percent
+--ro address-pool-status
    +--ro address-pool* [address-pool-name]
        +--ro address-pool-name      string
        +--ro address-pool-service* [service-name]
        |   +--ro service-name      string
        +--ro status?                enumeration
        +--ro address-pool-entries
            +--ro ipv4-address-range* [ipv4-address-range-name]
                |   +--ro ipv4-address-range-name      string
                |   +--ro peak-address-usage-ratio?    percent
                |   +--ro average-address-usage-ratio? percent
            +--ro ipv6-prefix* [ipv6-prefix-name]
                |   +--ro ipv6-prefix-name              string
                |   +--ro peak-prefix-usage-ratio?      percent
                |   +--ro average-prefix-usage-ratio?    percent
            +--ro port-range* [port-range-name]
                +--ro port-range-name      string
                +--ro peak-address-usage-ratio?    percent
                +--ro average-address-usage-ratio? percent

```

Figure 1: Interface to Address Pool Management (APM)

#### 4. AddressPoolManagement YANG Module

This module imports typedefs from [[RFC6991](#)] and [[RFC7223](#)].

<CODE BEGINS> file "ietf-address-pool@2015-10-14.yang"



```
module ietf-address-pool {
  namespace "urn:ietf:params:xml:ns:yang:ietf-address-pool";
  prefix address-pool;
  import ietf-inet-types {
    prefix inet;
  }
  import ietf-yang-types {
    prefix yang;
  }
  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
    WG List:  <mailto:netmod@ietf.org>

    WG Chair: Thomas Nadeau
    <mailto:tnadeau@lucidvision.com>

    WG Chair: Juergen Schoenwaelder
    <mailto:j.schoenwaelder@jacobs-university.de>

    Editor:   Qiong Sun
              <mailto:sunqiong@ctbri.com.cn>

    Editor:   Tao Peng
              <mailto:dr.pengtao@huawei.com>;

    Contributor: Will(Shucheng) Liu
              <mailto:liushucheng@huawei.com>";

  description
    "This module contains a collection of YANG definitions for
    configuring IP address pools.

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

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Simplified BSD License
    set forth in Section 4.c of the IETF Trust's Legal Provisions
    Relating to IETF Documents
    (http://trustee.ietf.org/license-info).
    This version of this YANG module is part of RFC 7277; see
    the RFC itself for full legal notices.";

  revision 2015-10-14 {
```





```
    description
        "Initial revision.";
    reference
        "RFC 7277: A YANG Data Model for IP Management";
}

typedef percent {
    type uint8 {
        range "0 .. 100";
    }
    description
        "Percentage";
}

typedef address-pool-type{
    type enumeration{
        enum usergateway {
            description
                "The address pool has a usergateway.";
        }
        enum import-route {
            description
                "The address pool need to import a route
                to external network.";
        }
    }
    description
        "Address pool type.";
}

typedef instance-type{
    type enumeration{
        enum pppoe {
            description
                "The address pool is used for pppoe access.";
        }
        enum dhcp {
            description
                "The address pool is used for dhcp access.";
        }
        enum vpn {
            description
                "The address pool is used for vpn access.";
        }
        enum ds-lite {
            description
                "The address pool is used for ds-lite access.";
        }
    }
}
```



```
    enum lw4over6 {
        description
            "The address pool is used for lw4over6 access.";
    }
    enum map {
        description
            "The address pool is used for map access.";
    }
    enum cgn {
        description
            "The address pool is used for cgn access.";
    }
    enum xlat {
        description
            "The address pool is used for xlat access.";
    }
    enum other {
        description
            "The address pool is used for others.";
    }
}
description
    "Instance type.";
}

container address-pools {
    description
        "This is a top level container for Address Pools.
        It can have one or more Address Pools. The pools may
        not be contiguous.";
    list address-pool {
        key address-pool-name;
        description
            "An Address Pool is an ordered list of
            Address Pool Entries (APE). Each Access Pool Entry has a
            list of address ranges and its associated lifetime.";
        leaf address-pool-name {
            type string;
            description
                "The name of address pool";
        }
        leaf address-pool-id {
            type uint32;
            default "0";
            description
                "The Address Pool id";
        }
    }
    leaf domain-name {
```



```
    type string;
    description
        "The domain name";
}
leaf device-id {
    type string;
    description
        "The identifier of device that using address pool";
}
list address-pool-service {
    key service-name;
    description
        "The services that can use these pool.";
    leaf service-name {
        type string;
        description
            "A service name: e.g., any, voip, iptv, internet, etc.";
    }
}

container address-pool-entries {
    description
        "The address-pool-entries container contains
        a list of address-ranges and associated attributes.";
    list ipv4-address-range {
        key ipv4-address-range-name;
        description
            "IPv4 Address range.";
        leaf ipv4-address-range-name {
            type string;
            description
                "The name of IPv4 address range.";
        }
        leaf ip-lower-address {
            type inet:ipv4-address-no-zone;
            description
                "The lower IPv4 address of the address range.";
        }
        leaf ip-upper-address {
            type inet:ipv4-address-no-zone;
            description
                "The upper IPv4 address of the address range.";
        }
        leaf usergateway {
            type inet:ipv4-address-no-zone;
            description
                "It only exists when address pool are used for
                user addressing.";
        }
    }
}
```



```
    }
    leaf gwnetmask {
        type yang:dotted-quad;
        description
            "The netmask for usergateway.";
    }
    leaf type {
        type address-pool-type;
        description
            "The type of the address pool.";
    }
    leaf lifetime {
        type yang:date-and-time;
        description
            "The lifetime for the address pool. '0' means
withdrawal.";
    }
    leaf instance {
        type instance-type;
        description
            "The instance of the address pool.";
    }
}
leaf warning-threshold-v4{
    type percent;
    description
        "The threshold of the ipv4 address pool.";
}

list ipv6-prefix {
    key ipv6-prefix-name;
    description
        "IPv6 prefix.";
    leaf ipv6-prefix-name {
        type string;
        description
            "The name of IPv6 prefix.";
    }
    leaf ipv6-prefix {
        type inet:ipv6-prefix;
        description
            "The IPv6 prefix.";
    }
}
leaf usergateway {
    type inet:ipv6-address-no-zone;
    description
        "It only exists when address pool are used for
user addressing.";
```





```
    }
    leaf type {
        type address-pool-type;
        description
            "The type of the address pool.";
    }
    leaf lifetime {
        type yang:date-and-time;
        description
            "The lifetime for the address pool. '0' means
withdrawal.";
    }
    leaf instance {
        type instance-type;
        description
            "The instance of the address pool.";
    }
}
leaf warning-threshold-v6{
    type percent;
    description
        "The threshold of the ipv6 address pool.";
}
}
}

/*
* Operational state data nodes
*/

container address-pool-status {
    config false;
    description
        "This is a top level container for Address Pool Status,
        which contains the status of address pool usage.";
    list address-pool {
        key address-pool-name;
        description
            "An Address Pool is an ordered list of
            Address Pool Entries (APE). Each Access Pool Entry has a
            list of address ranges and its associated lifetime. ";
        leaf address-pool-name {
            type string;
            description
                "The name of address pool";
        }
        list address-pool-service {
```



```
    key service-name;
    description
        "The services that can use these pool.";
    leaf service-name {
        type string;
        description
            "A service name: e.g., any, voip, iptv, internet, etc.";
    }
}
leaf status {
    type enumeration{
        enum active {
            description
                "The address pool is in active status.";
        }
        enum idle {
            description
                "The address pool is in idle status.";
        }
    }
    description
        "The status of address pool";
}
container address-pool-entries {
    description
        "The address-pool-entries container contains
        a list of address-ranges and associated attributes.";
    list ipv4-address-range {
        key ipv4-address-range-name;
        description
            "IPv4 Address range.";
        leaf ipv4-address-range-name {
            type string;
            description
                "The name of IPv4 address range.";
        }
        leaf peak-address-usage-ratio {
            type percent;
            description
                "The peak usage rate of the address range.";
        }
        leaf average-address-usage-ratio {
            type percent;
            description
                "The average usage rate of the address range.";
        }
    }
}
list ipv6-prefix {
```



```
key ipv6-prefix-name;
description
    "IPv6 prefix.";
leaf ipv6-prefix-name {
    type string;
    description
        "The name of IPv6 prefix.";
}
leaf peak-prefix-usage-ratio {
    type percent;
    description
        "The peak usage rate of the prefix.";
}
leaf average-prefix-usage-ratio {
    type percent;
    description
        "The average usage rate of the prefix.";
}
}
list port-range {
    key port-range-name;
    description
        "port range.";
    leaf port-range-name {
        type string;
        description
            "The name of port range.";
    }
    leaf peak-address-usage-ratio {
        type percent;
        description
            "The peak usage rate of the port range.";
    }
    leaf average-address-usage-ratio {
        type percent;
        description
            "The average usage rate of the port range.";
    }
}
}
}
}
<CODE ENDS>
```

Figure 2: Interface to Address Pool Management (APM)



## 5. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [[RFC6241](#)]. The lowest NETCONF layer is the secure transport layer and the support of SSH is mandatory to implement secure transport [[RFC6242](#)]. The NETCONF access control model [[RFC6536](#)] provides means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and contents.

All data nodes defined in the YANG module which can be created, modified and deleted (i.e., config true, which is the default). These data nodes are considered sensitive. Write operations (e.g., edit-config) applied to these data nodes without proper protection can negatively affect network operations.

## 6. IANA Considerations

This document requests IANA to register the following URI in the "IETF XML Registry" [[RFC3688](#)]:

```
URI: urn:ietf:params:xml:ns:yang:ietf-address-pool
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.
```

Figure 3: namespace

This document requests IANA to register the following YANG module in the "YANG Module Names" registry [[RFC6020](#)].

```
name: ietf-address-pool
namespace: urn:ietf:params:xml:ns:yang:ietf-address-pool
prefix: address-pool
reference: RFC XXXX
```

Figure 4: IANA register

## 7. Contributors

The following people contributed to creating this document, and are listed in alphabetical order:

Will(Shucheng) Liu





## **8. Acknowledgements**

N/A.

## **9. References**

### **9.1. 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>>.
- [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>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6991](#), DOI 10.17487/RFC6991, July 2013, <<http://www.rfc-editor.org/info/rfc6991>>.
- [RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", [RFC 7223](#), DOI 10.17487/RFC7223, May 2014, <<http://www.rfc-editor.org/info/rfc7223>>.

### **9.2. Informative References**

- [RFC6674] Brockners, F., Gundavelli, S., Speicher, S., and D. Ward, "Gateway-Initiated Dual-Stack Lite Deployment", [RFC 6674](#), DOI 10.17487/RFC6674, July 2012, <<http://www.rfc-editor.org/info/rfc6674>>.
- [RFC6888] Perreault, S., Ed., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", [BCP 127](#), [RFC 6888](#), DOI 10.17487/RFC6888, April 2013, <<http://www.rfc-editor.org/info/rfc6888>>.

#### **Authors' Addresses**

Qiong Sun  
China Telecom  
No.118 Xizhimennei street, Xicheng District  
Beijing 100035  
P.R. China

Email: [sunqiong@ctbri.com.cn](mailto:sunqiong@ctbri.com.cn)



Chongfeng Xie  
China Telecom  
No.118 Xizhimennei street, Xicheng District  
Beijing 100035  
P.R. China

Email: xiechf@ctbri.com.cn

Mohamed Boucadair  
Orange  
Rennes 35000  
France

Email: mohamed.boucadair@orange-ftgroup.com

Tao Peng  
Huawei Technologies  
Bantian, Longgang District  
Shenzhen 518129  
China

Email: dr.pengtao@huawei.com

Yiu L. Lee  
Comcast  
One Comcast Center  
Philadelphia, PA 19103  
USA

Email: yiu\_lee@cable.comcast.com

