Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 17, 2013

M. Bjorklund
Tail-f Systems

July 16, 2012

A YANG Data Model for IP Configuration
draft-ietf-netmod-ip-cfg-05

Abstract

This document defines a YANG data model for configuration of IP implementations.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

This document defines a YANG [RFC6020] data model for configuration of IP implementations.

The initial version of this data model focuses on configuration parameters for interfaces. Future revisions of this data model might add other kinds of IP configuration parameters.

Configuration parameters to control IP routing are defined in [I-D.ietf-netmod-routing-cfg].

1.1. Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, [RFC2119].

The following terms are defined in [RFC6241] and are not redefined here:

- client

- server

The following terms are defined in [RFC6020] and are not redefined here:

- augment

- data model

- data node

2. IP Data Model

The module "ietf-ip" augments the "interface" list defined in the "ietf-interfaces" module [I-D.ietf-netmod-interfaces-cfg] with the following data nodes, where square brackets are used to enclose a list’s keys, and "?" means that the node is optional. Choice and case nodes are enclosed in parenthesis, and a case node is marked with a colon (":").

```
      +--rw if:interfaces
         +--rw if:interface [name]
            ...
            +--rw ipv4
            |  +--rw enabled?            boolean
            |  +--rw ip-forwarding?      boolean
            |  +--rw address [ip]
            |  |  +--rw ip                inet:ipv4-address
            |  |  +--rw (subnet)?
            |  |     +--:(prefix-length)
            |  |     |  +--rw ip:prefix-length?   uint8
            |  |     +--:(netmask)
            |  |        +--rw ip:netmask?         inet:ipv4-address
            |  +--rw neighbor [ip]
            |     +--rw ip                inet:ipv4-address
            |     +--rw phys-address?    yang:phys-address
            +--rw ipv6
               +--rw enabled?            boolean
               +--rw ip-forwarding?      boolean
               +--rw address [ip]
               |  +--rw ip                inet:ipv6-address
               |  +--rw prefix-length?   uint8
               +--rw neighbor [ip]
               |  +--rw ip                inet:ipv6-address
               |  +--rw phys-address?    yang:phys-address
               +--rw dup-addr-detect-transmits?   uint32
               +--rw autoconf
                  +--rw create-global-addresses?        boolean
                  +--rw create-temporary-addresses?     boolean
                  +--rw temporary-valid-lifetime?       uint32
                  +--rw temporary-preferred-lifetime?   uint32
```

The data model defines two containers, "ipv4" and "ipv6", representing the IPv4 and IPv6 address families. In each container, there is a leaf "enabled" that controls if the address family is enabled on that interface, and a leaf "ip-forwarding" that controls if ip packet forwarding for the address family is enabled on the interface. In each container, there is also a list of manually configured addresses, and a list of manually configured mappings from ip addresses to physical addresses.

3. Relationship to IP-MIB

If the device implements IP-MIB [RFC4293], each entry in the "ipv4/address" and "ipv6/address" lists is mapped to one ipAddressEntry, where the ipAddressIfIndex refers to the interface where the "address" entry is configured.

The IP-MIB defines objects to control IPv6 Router Advertisement. The corresponding YANG data nodes are defined in [I-D.ietf-netmod-routing-cfg].

The entries in "ipv4/neighbor" and "ipv6/neighbor" are mapped to ipNetToPhysicalTable.

The object ipAddressStatus is writable in the IP-MIB but does not represent configuration, and is thus not mapped to the YANG module.

The following table lists the YANG data nodes with corresponding objects in the IP-MIB.

YANG data node	IP-MIB object
ipv4/enabled	ipv4InterfaceEnableStatus
ipv4/address	ipAddressEntry
ipv4/address/ip	ipAddressAddrType / ipAddressAddr
ipv4/neighbor	ipNetToPhysicalTable
ipv6/enabled	ipv6InterfaceEnableStatus
ipv6/ip-forwarding	ipv6InterfaceForwarding
ipv6/address	ipAddressEntry
ipv6/address/ip	ipAddressAddrType / ipAddressAddr
ipv6/neighbor	ipNetToPhysicalTable

Mapping of YANG data nodes to IP-MIB objects

4. IP configuration YANG Module

This module imports typedefs from [RFC6021] and [I-D.ietf-netmod-interfaces-cfg], and references [RFC0826], [RFC4861] and [RFC4862].

RFC Ed.: update the date below with the date of RFC publication and remove this note.

```
<CODE BEGINS> file "ietf-ip@2012-07-16.yang"

module ietf-ip {

  namespace "urn:ietf:params:xml:ns:yang:ietf-ip";
  prefix ip;

  import ietf-interfaces {
    prefix if;
  }
  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: David Kessens
               <mailto:david.kessens@nsn.com>

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

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>";

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

     Copyright (c) 2012 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 XXXX; see
        the RFC itself for full legal notices.";

     // RFC Ed.: replace XXXX with actual RFC number and remove this
     // note.

     // RFC Ed.: update the date below with the date of RFC publication
     // and remove this note.
     revision 2012-07-16 {
       description
         "Initial revision.";
       reference
         "RFC XXXX: A YANG Data Model for IP Configuration";
     }

     /* Features */

     feature non-contiguous-netmasks {
       description
         "Indicates support for configuring non-contiguous
          subnet masks.";
     }

     /* Data nodes */

     augment "/if:interfaces/if:interface" {
       description
         "Parameters for configuring IP on interfaces.

          If an interface is not capable of running IP, the server
          must not allow the client to configure these parameters.";

       container ipv4 {
         description
           "Parameters for the IPv4 address family.";
         leaf enabled {
           type boolean;
           default true;
           description
             "Controls if IPv4 is enabled or disabled on this
              interface.";
         }
         leaf ip-forwarding {
           type boolean;
           default false;
           description
             "Controls if IPv4 packet forwarding is enabled or disabled
              on this interface.";
         }
         list address {
           key "ip";
           description
             "The list of manually configured IPv4 addresses
              on the interface.";

           leaf ip {
             type inet:ipv4-address;
             description
               "The IPv4 address on the interface.";
           }
           choice subnet {
             default prefix-length;
             description
               "The subnet can be specified as a prefix-length, or,
                if the server supports non-contiguous netmasks, as
                a netmask.

                The default subnet is a prefix-length of 32.";
             leaf prefix-length {
               type uint8 {
                 range "0..32";
               }
               default 32;
               description
                 "The length of the subnet prefix.";
             }
             leaf netmask {
               if-feature non-contiguous-netmasks;
               type inet:ipv4-address;
               description
                 "The subnet specified as a netmask.";
             }
           }
         }
         list neighbor {
           key "ip";
           description
             "A list of manually configured mappings from IPv4
              addresses to physical addresses.
              Entries in this list are used as static entries in the
              ARP cache.";
           reference
             "RFC 826: An Ethernet Address Resolution Protocol";

           leaf ip {
             type inet:ipv4-address;
             description
               "The IPv4 address of a neighbor node.";
           }
           leaf phys-address {
             type yang:phys-address;
             description
               "The physical level address of the neihgbor node.";
           }
         }

       }
       container ipv6 {
         description
           "Parameters for the IPv6 address family.";

         leaf enabled {
           type boolean;
           default true;
           description
             "Controls if IPv6 is enabled or disabled on this
              interface.";
         }
         leaf ip-forwarding {
           type boolean;
           default false;
           description
             "Controls if IPv6 packet forwarding is enabled or disabled
              on this interface.";
           reference
             "RFC 4861: Neighbor Discovery for IP version 6 (IPv6)
                        Section 6.2.1, IsRouter";
         }
         list address {
           key "ip";
           description
             "The list of manually configured IPv6 addresses
              on the interface.";

           leaf ip {
             type inet:ipv6-address;
             description
```

```
               "The IPv6 address on the interface.";
           }
           leaf prefix-length {
             type uint8 {
               range "0..128";
             }
             default 128;
             description
               "The length of the subnet prefix.";
           }
         }
         list neighbor {
           key "ip";
           description
             "A list of manually configured mappings from IPv6
              addresses to physical addresses.

              Entries in this list are used as static entries in the
              Neighbor Cache.";
           reference
             "RFC 4861: Neighbor Discovery for IP version 6 (IPv6)";

           leaf ip {
             type inet:ipv6-address;
             description
               "The IPv6 address of a neighbor node.";
           }
           leaf phys-address {
             type yang:phys-address;
             description
               "The physical level address of the neihgbor node.";
           }
         }
         leaf dup-addr-detect-transmits {
           type uint32;
           default 1;
           description
             "The number of consecutive Neighbor Solicitation messages
              sent while performing Duplicate Address Detection on a
              tentative address.  A value of zero indicates that
              Duplicate Address Detection is not performed on
              tentative addresses.  A value of one indicates a single
              transmission with no follow-up retransmissions.";
           reference
             "RFC 4862: IPv6 Stateless Address Autoconfiguration";
         }
         container autoconf {
           description
```

```
             "Parameters to control the autoconfiguration of IPv6
              addresses, as described in RFC 4862.";
           reference
             "RFC 4862: IPv6 Stateless Address Autoconfiguration";

           leaf create-global-addresses {
             type boolean;
             default true;
             description
               "If enabled, the host creates global addresses as
                described in section 5.5 of RFC 4862.";
             reference
               "RFC 4862: IPv6 Stateless Address Autoconfiguration";
           }
           leaf create-temporary-addresses {
             type boolean;
             default false;
             description
               "If enabled, the host creates temporary addresses as
                described in RFC 4941.";
             reference
               "RFC 4941: Privacy Extensions for Stateless Address
                          Autoconfiguration in IPv6";
           }
           leaf temporary-valid-lifetime {
             type uint32;
             units "seconds";
             default 604800;
             description
               "The time the temporary address is valid.";
             reference
               "RFC 4941: Privacy Extensions for Stateless Address
                          Autoconfiguration in IPv6
                          - TEMP_VALID_LIFETIME";
           }
           leaf temporary-preferred-lifetime {
             type uint32;
             units "seconds";
             default 86400;
             description
               "The time the temporary address is preferred.";
             reference
               "RFC 4941: Privacy Extensions for Stateless Address
                          Autoconfiguration in IPv6
                          - TEMP_PREFERED_LIFETIME";
           }
         }
       }
```

```
     }
   }

   <CODE ENDS>
```

5. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in RFC 3688, the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-ip

Registrant Contact: The NETMOD WG of the IETF.

XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

name:	ietf-ip
namespace:	urn:ietf:params:xml:ns:yang:ietf-ip
prefix:	ip
reference:	RFC XXXX

6. 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 mandatory-to-implement secure transport is SSH [RFC6242].

There are a number of data nodes defined in the YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

ipv4/enabled and ipv6/enabled: These leafs are used to enable or disable IPv4 and IPv6 on a specific interface. By enabling a protocol on an interface, an attacker might be able to create an unsecured path into a node (or through it if routing is also enabled). By disabling a protocol on an interface, an attacker might be able to force packets to be routed through some other interface or deny access to some or all of the network via that protocol.

ipv4/address and ipv6/address: These lists specify the configured IP addresses on an interface. By modifying this information, an attacker can cause a node to either ignore messages destined to it or accept (at least at the IP layer) messages it would otherwise ignore. The use of filtering or security associations may reduce the potential damage in the latter case.

ipv4/ip-forwarding and ipv6/ip-forwarding: These leafs allow a client to enable or disable the routing functions on the entity. By disabling the routing functions, an attacker would possibly be able to deny service to users. By enabling the routing functions, an attacker could open a conduit into an area. This might result in the area providing transit for packets it shouldn’t or might allow the attacker access to the area bypassing security safeguards. =ipv6/autoconf: The leafs in this branch control the autoconfiguration of IPv6 addresses and in particular whether temporary addresses are used or not. By modifying the corresponding leafs, an attacker might impact the addresses used by a node and thus indirectly the privacy of the users using the node.

7. Acknowledgments

The author wishes to thank Ladislav Lhotka, Juergen Schoenwaelder, and Dave Thaler for their helpful comments.

8. References

8.1. Normative References

[I-D.ietf-netmod-interfaces-cfg]
Bjorklund, M., "A YANG Data Model for Interface Configuration", draft-ietf-netmod-interfaces-cfg-05 (work in progress), July 2012.

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

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

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

[RFC6021] Schoenwaelder, J., "Common YANG Data Types", RFC 6021, October 2010.

8.2. Informative References

[I-D.ietf-netmod-routing-cfg]
Lhotka, L., "A YANG Data Model for Routing Configuration", draft-ietf-netmod-routing-cfg-04 (work in progress), July 2012.

[RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware", STD 37, RFC 826, November 1982.

[RFC4293] Routhier, S., "Management Information Base for the Internet Protocol (IP)", RFC 4293, April 2006.

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

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.

Appendix A. Example: NETCONF <get> reply

This section gives an example of a reply to the NETCONF <get> request for a device that implements the data model defined in this document.

```
<rpc-reply
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
    message-id="101">
  <data>
    <interfaces
        xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
      <interface>
        <name>eth0</name>
        <type>ethernetCsmacd</type>
        <location>0</location>
        <if-index>2</if-index>
        <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
          <address>
            <ip>192.0.2.1</ip>
            <prefix-length>24</prefix-length>
          </address>
        </ipv4>
        <ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
          <address>
            <ip>2001:DB8::1</ip>
            <prefix-length>32</prefix-length>
          </address>
          <dup-addr-detect-transmits>0</dup-addr-detect-transmits>
        </ipv6>
      </interface>
    </interfaces>
  </data>
</rpc-reply>
```

Author's Address

Martin Bjorklund
Tail-f Systems

Email: mbj@tail-f.com