Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 11, 2012

M. Bjorklund
Tail-f Systems

February 8, 2012

A YANG Data Model for Interface Configuration
draft-ietf-netmod-interfaces-cfg-03

Abstract

This document defines a YANG data model for the configuration of network interfaces. It is expected that interface type specific configuration data models augment the generic interfaces data model defined in this document.

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 August 11, 2012.

Table of Contents

1. Introduction

This document defines a YANG [RFC6020] data model for the configuration of network interfaces. It is expected that interface type specific configuration data models augment the generic interfaces data model defined in this document.

Network interfaces are central to the configuration of many Internet protocols. Thus, it is important to establish a common data model for how interfaces are identified and configured.

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].

2. Objectives

This section describes some of the design objectives for the model presented in Section 5.

- It is recognized that existing implementations will have to map the interface data model defined in this memo to their proprietary native data model. The new data model should be simple to facilitate such mappings.

- The data model should be suitable for new implementations to use as-is, without requiring a mapping to a different native model.

- References to interfaces should be as simple as possible, preferably by using a single leafref.

- The mapping to ifIndex [RFC2863] used by SNMP to identify interfaces must be clear.

- The model must support interface layering, both simple layering where one interface is layered on top of exactly one other interface, and more complex scenarios where one interface is aggregated over N other interfaces, or when N interfaces are multiplexed over one other interface.

- The data model should support the pre-provisioning of interface configuration, i.e., it should be possible to configure an interface whose physical interface hardware is not present on the device. It is recommended that devices that support dynamic addition and removal of physical interfaces also support pre-provisioning.

3. Interfaces Data Model

The data model in the module "ietf-interfaces" has the following structure, where square brackets are used to enclose a list's keys, and "?" means that the leaf is optional:

```
   +--rw interfaces
      +--rw interface [name]
         +--rw name                        string
         +--rw description?                string
         +--rw type                        ianaift:iana-if-type
         +--rw location?                   string
         +--rw enabled?                    boolean
         +--ro if-index                    int32
         +--rw mtu?                        uint32
         +--rw link-up-down-trap-enable?   enumeration
```

This module defines one YANG feature:

snmp-if-mib: Indicates that the server implements IF-MIB [RFC2863].

3.1. The interface List

The data model for interface configuration presented in this document uses a flat list of interfaces. Each interface in the list is identified by its name. Furthermore, each interface has a mandatory "type" leaf, and a "location" leaf. The combination of "type" and "location" is unique within the interface list.

It is expected that interface type specific data models augment the interface list, and use the "type" leaf to make the augmentation conditional.

As an example of such an interface type specific augmentation, consider this YANG snippet. For a more complete example, see Appendix A.

```
import interfaces {
    prefix "if";
}

augment "/if:interfaces/if:interface" {
    when "if:type = 'ethernetCsmacd'";

    container ethernet {
        leaf duplex {
            ...
        }
    }
}
```

The "location" leaf is a string. It is optional in the data model, but if the type represents a physical interface, it is mandatory. The format of this string is device- and type-dependent. The device uses the location string to identify the physical or logical entity that the configuration applies to. For example, if a device has a single array of 8 ethernet ports, the location can be one of the strings "1" to "8". As another example, if a device has N cards of M ports, the location can be on the form "n/m", such as "1/0".

How a client can learn which types and locations are present on a certain device is outside the scope of this document.

3.2. Interface References

An interface is uniquely identified by its name. This property is captured in the "interface-ref" typedef, which other YANG modules SHOULD use when they need to reference an existing interface.

3.3. Interface Layering

There is no generic mechanism for how an interface is configured to be layered on top of some other interface. It is expected that interface type specific models define their own nodes for interface layering, by using "interface-ref" types to reference lower layers.

Below is an example of a model with such nodes. For a more complete example, see Appendix B.

```
     augment "/if:interfaces/if:interface" {
         when "if:type = 'ieee8023adLag'";

         leaf-list slave-if {
             type if:interface-ref;
             must "/if:interfaces/if:interface[if:name = current()]"
                + "/if:type = 'ethernetCsmacd'" {
                 description
                     "The type of a slave interface must be ethernet";
             }
         }
         // other bonding config params, failover times etc.
     }
```

4. Relationship to the IF-MIB

If the device implements IF-MIB [RFC2863], each entry in the "interface" list is typically mapped to one ifEntry. The "if-index" leaf contains the value of the corresponding ifEntry's ifIndex.

In most cases, the "name" of an "interface" entry is mapped to ifName. ifName is defined as an DisplayString [RFC2579] which uses a 7-bit ASCII character set. An implementation MAY restrict the allowed values for "name" to match the restrictions of ifName.

The IF-MIB allows two different ifEntries to have the same ifName. Devices that support this feature, and also support the configuration of these interfaces using the "interface" list, cannot have a 1-1 mapping between the "name" leaf and ifName.

5. Interfaces YANG Module

This YANG module imports a typedef from
[I-D.ietf-netmod-iana-if-type].

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

<CODE BEGINS> file "ietf-interfaces@2012-02-08.yang"

```
module ietf-interfaces {

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

  import iana-if-type {
    prefix ianaift;
  }

  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 network interfaces.

     Copyright (c) 2011 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 2012-02-08 {
        description
          "Initial revision.";
        reference
          "RFC XXXX: A YANG Data Model for Interface Configuration";
      }

      /* Typedefs */

      typedef interface-ref {
        type leafref {
          path "/if:interfaces/if:interface/if:name";
        }
        description
          "This type is used by data models that need to reference
           interfaces.";
      }

      /* Features */

      feature snmp-if-mib {
        description
          "This feature indicates that the server implements IF-MIB.";
        reference
          "RFC 2863: The Interfaces Group MIB";
      }

      /* Data nodes */

      container interfaces {
        description
          "Interface parameters.";

        list interface {
          key "name";
          unique "type location";

          description
            "The list of configured interfaces on the device.";
```

```
        leaf name {
          type string;
          description
            "An arbitrary name for the interface.

             A device MAY restrict the allowed values for this leaf,
             possibly depending on the type and location.

             For example, if a device has a single array of 8 ethernet
             ports, the name might be restricted to be on the form
            'ethN', where N is an integer between '1' and '8'.

             This leaf MAY be mapped to ifName by an implementation.
             Such an implementation MAY restrict the allowed values
             for this leaf so that it matches the restrictions of
             ifName.";
          reference
            "RFC 2863: The Interfaces Group MIB - ifName";
        }

        leaf description {
          type string;
          description
            "A textual description of the interface.

             This leaf MAY be mapped to ifAlias by an implementation.
             Such an implementation MAY restrict the allowed values
             for this leaf so that it matches the restrictions of
             ifAlias.";
          reference
            "RFC 2863: The Interfaces Group MIB - ifAlias";
        }

        leaf type {
          type ianaift:iana-if-type;
          mandatory true;
          description
            "The type of the interface.

             When an interface entry is created, a server MAY
             initialize the type leaf with a valid value, e.g., if it
             is possible to derive the type from the name of the
             interface.";
        }

        leaf location {
          type string;
          description
```

```
          "The device-specific location of the interface of a
           particular type.  The format of the location string
           depends on the interface type and the device.

           Media-specific modules must specify if the location
           is needed for the given type.

           For example, if a device has a single array of 8 ethernet
           ports, the location can be one of '1' to '8'.  As another
           example, if a device has N cards of M ports, the location
           can be on the form 'n/m'.

           When an interface entry is created, a server MAY
           initialize the location leaf with a valid value, e.g., if
           it is possible to derive the location from the name of
           the interface.";
      }

      leaf enabled {
        type boolean;
        default "true";
        description
          "The desired state of the interface.

           This leaf contains the configured, desired state of the
           interface.  Systems that implement the IF-MIB use the
           value of this leaf to set IF-MIB.ifAdminStatus to 'up' or
           'down' after an ifEntry has been initialized, as described
           in RFC 2863.";
        reference
          "RFC 2863: The Interfaces Group MIB - ifAdminStatus";
      }

      leaf if-index {
        if-feature snmp-if-mib;
        type int32 {
          range "1..2147483647";
        }
        config false;
        description
          "The ifIndex value for the ifEntry represented by this
           interface.

           Media-specific modules must specify how the type is
           mapped to entries in the ifTable.";
        reference
          "RFC 2863: The Interfaces Group MIB - ifIndex";
      }
```

```
        leaf mtu {
          type uint32;
          description
            "The size, in octets, of the largest packet that the
             interface can send and receive.  This node might not be
             valid for all interface types.

             Media-specific modules must specify any restrictions on
             the mtu for their interface type.";
        }

        leaf link-up-down-trap-enable {
          if-feature snmp-if-mib;
          type enumeration {
            enum enabled {
              value 1;
            }
            enum disabled {
              value 2;
            }
          }
          description
            "Indicates whether linkUp/linkDown SNMP notifications
             should be generated for this interface.

             If this node is not configured, the value 'enabled' is
             operationally used by the server for interfaces which do
             not operate on top of any other interface (as defined in
             the ifStackTable), and 'disabled' otherwise.";
          reference
            "RFC 2863: The Interfaces Group MIB -
                       ifLinkUpDownTrapEnable";
        }
      }
    }
  }

  <CODE ENDS>
```

6. 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-interfaces

Registrant Contact: The IESG.

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-interfaces
namespace:    urn:ietf:params:xml:ns:yang:ietf-interfaces
prefix:       if
reference:    RFC XXXX
```

7. 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:

/interfaces/interface: This list specify the configured interfaces on a device. Unauthorized access to this list could cause the device to ignore packets destined to it.

/interfaces/interface/enabled: This leaf controls if an interface is enabled or not. Unauthorized access to this leaf could cause the device to ignore packets destined to it.

8. Acknowledgments

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

9. Normative References

[I-D.ietf-netmod-iana-if-type]
Bjorklund, M., "IANA Interface Type YANG Module", draft-ietf-netmod-iana-if-type-00 (work in progress), April 2011.

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

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

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

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

[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: Ethernet Interface Module

This section gives a simple example of how an Ethernet interface module could be defined. It demonstrates how media-specific configuration parameters can be conditionally augmented to the generic interface list. It is not intended as a complete module for ethernet configuration.

```
module ex-ethernet {
  namespace "http://example.com/ethernet";
  prefix "eth";

  import ietf-interfaces {
    prefix if;
  }

  augment "/if:interfaces/if:interface" {
    when "if:type = 'ethernetCsmacd'";

    container ethernet {
      must "../if:location" {
        description
          "An ethernet interface must specify the physical location
           of the ethernet hardware.";
      }
      choice transmission-params {
        case auto {
          leaf auto-negotiate {
            type empty;
          }
        }
        case manual {
          leaf duplex {
            type enumeration {
              enum "half";
              enum "full";
            }
          }
          leaf speed {
            type enumeration {
              enum "10Mb";
              enum "100Mb";
              enum "1Gb";
              enum "10Gb";
            }
          }
        }
      }
      // other ethernet specific params...
    }
  }
}
```

Appendix B. Example: Ethernet Bonding Interface Module

This section gives an example of how interface layering can be defined. An ethernet bonding interface is defined, which bonds several ethernet interfaces into one logical interface.

```
module ex-ethernet-bonding {
  namespace "http://example.com/ethernet-bonding";
  prefix "bond";

  import ietf-interfaces {
    prefix if;
  }

  augment "/if:interfaces/if:interface" {
    when "if:type = 'ieee8023adLag'";

    leaf-list slave-if {
      type if:interface-ref;
      must "/if:interfaces/if:interface[if:name = current()]"
         + "/if:type = 'ethernetCsmacd'" {
        description
          "The type of a slave interface must be ethernet.";
      }
    }
    leaf bonding-mode {
      type enumeration {
        enum round-robin;
        enum active-backup;
        enum broadcast;
      }
    }
    // other bonding config params, failover times etc.
  }
}
```

Appendix C. Example: VLAN Interface Module

This section gives an example of how a vlan interface module can be defined.

```
module ex-vlan {
  namespace "http://example.com/vlan";
  prefix "vlan";

  import ietf-interfaces {
    prefix if;
  }

  augment "/if:interfaces/if:interface" {
    when "if:type = 'ethernetCsmacd' or
          if:type = 'ieee8023adLag'";
    leaf vlan-tagging {
      type boolean;
      default false;
    }
  }

  augment "/if:interfaces/if:interface" {
    when "if:type = 'l2vlan'";

    leaf base-interface {
      type if:interface-ref;
      must "/if:interfaces/if:interface[if:name = current()]"
         + "/vlan:vlan-tagging = true" {
        description
          "The base interface must have vlan tagging enabled.";
      }
    }
    leaf vlan-id {
      type uint16 {
        range "1..4094";
      }
      must "../base-interface";
    }
  }
}
```

Appendix D. Example: NETCONF <get> reply

This section gives an example of a reply to the NETCONF <get> request for a device that implements the example data models above.

```
<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>
        <enabled>true</enabled>
        <if-index>2</if-index>
      </interface>
      <interface>
        <name>eth1</name>
        <type>ethernetCsmacd</type>
        <location>1</location>
        <enabled>true</enabled>
        <if-index>7</if-index>
        <vlan-tagging
            xmlns="http://example.com/vlan">true</vlan-tagging>
      </interface>
    </interfaces>
  </data>
</rpc-reply>
```

Appendix E. ChangeLog

RFC Editor: remove this section upon publication as an RFC.

E.1. Version -03

- Added the section Relationship to the IF-MIB.
- Changed if-index to be a leaf instead of leaf-list.
- Explained the notation used in the data model tree picture.

E.2. Version -02

- Editorial fixes

E.3. Version -01

- Changed leaf "if-admin-status" to leaf "enabled".
- Added Security Considerations

Author's Address

Martin Bjorklund
Tail-f Systems

Email: mbj@tail-f.com