

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 4, 2017

Y. Liu
A. Foldes
Ericsson
G. Zheng
Z. Wang
Y. Zhuang
Huawei
A. Wang
China Telecom
July 3, 2016

Yang Data Model for IPIPv4 Tunnel
draft-liu-intarea-ipipv4-tunnel-yang-02

Abstract

This document defines a YANG data model for the management of IP based tunnels. The data model includes configuration data and state data. In default format, it describes managed attributes used for managing tunnels of IPv4 or IPv6 over IPv4 tunnels. And it can also serve as a base model which can be augmented with technology-specific details in other Ip based tunnel models.

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 January 4, 2017.

Copyright Notice

Copyright (c) 2016 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
1.2. Tree Diagrams	3
2. Architecture of IP tunnel YANG Model	3
3. IP Tunnel Model Design	4
3.1. Model Overview	4
3.2. IP Tunnel YANG Tree Diagrams	5
4. IP Tunnel YANG Data Model	6
5. Security Considerations	13
6. IANA Considerations	13
7. Normative References	14
Authors' Addresses	14

1. Introduction

An overview of tunnel is presented at [intarea-tunnel]. Over the past several years, there has been a number of "tunneling" protocols specified by the IETF. And this document defines a YANG data model for the management of IP bases tunnels. In default format, it covers the following tunnel types:

- o IPv4 in IPv4, related concepts are defined in [RFC1853]
- o IPv6 in IPv4 manual tunnel, related concepts are defined in [RFC2003]
- o IPv6 to IPv4 tunnel, related concepts are defined in [RFC3056]

And notice that this model provides a framework and some reusable common attributes where technology-specific IP tunnel YANG models can inherit constructs from it without needing to redefine them within the sub-technology. Therefore it also can serve as a base model which can be extended to include technology specific details.

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].

1.2. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is as follows:

- o Brackets "[" and "]" enclose list keys.
- o Abbreviations before data node names: "rw" means configuration (read-write), and "ro" means state data (read-only).
- o Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.
- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

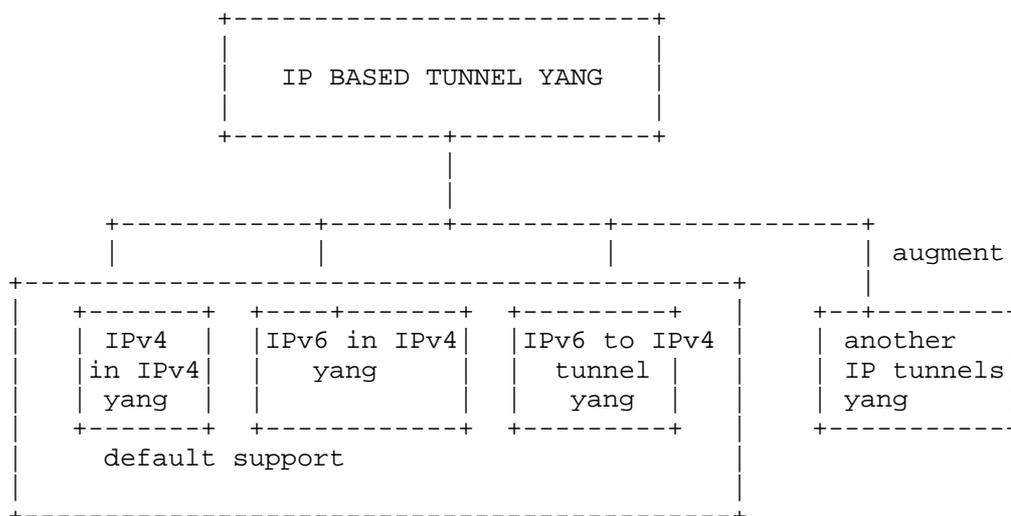
2. Architecture of IP tunnel YANG Model

In this document we define an IP based tunnel model, in default it can describe the IPv6/4-in-IPv4 tunnels including IPv4 in IPv4 [RFC1853], IPv6 in IPv4 [RFC2003], IPv6 to IPv4 [RFC3056].

Since some reusable common core attributes of ip tunnels are defined, it can also provide an optional solution for extending to the other IP-based tunnel models. The users of IP-based tunnel model can according to following method to define other technologies specific ip tunnel models:

- o The IP based Tunnel YANG model can acts as the root for other Tunnel YANG models.
- o Augment the ip based tunnel model by adding the tunnel type is defined as an identity that augments the base " ip-tunnel-type " defined in the IP based model.
- o Augment the ip based tunnel model by adding new data nodes with technology specific parameters into proper anchor points of the ip tunnel model.

Figure 1 depicts the relationship of different Tunnel YANG models to the Ip Tunnel YANG Model. It can default support the IPv4-in-IPv4, IPv6-in-IPv4, and IPv6-to-IPv4, and can also be extended to another IP base tunnels encapsulation protocol.



Relationship of technology specific TUNNEL YANG model to IP based tunnel YANG model

3. IP Tunnel Model Design

3.1. Model Overview

This document defines the YANG model "ietf-ip-tunnel". It includes two modules, one for configuration and one for state. At the top of the Model there is Tunnels container for tunnel configuration data. Multiple Tunnel lists keyed using tunnel name and tunnel type within the Tunnels container. To correctly identify an ip based tunnel the bind-interface, local-address, remote-address are defined under one tunnel list. Notice that tunnels are handled by creating a logical interface for each tunnel. Each logical interface (physical or virtual) need to map to interface yang model [RFC7223]. To do so, in this draft, the bind-interface are defined which with a leafref type and it can be used pointing to the corresponding interface-name node in the interface yang [RFC7223].

The data model also includes read-only counter so that the tunnel state can be read.

3.2. IP Tunnel YANG Tree Diagrams

The data model has the following tree diagram for the IP tunnels:

```

module: ietf-ip-tunnel
  +--rw Tunnels
  |   +--rw Tunnel* [name type]
  |   |   +--rw name                string
  |   |   +--rw type                identityref
  |   |   +--rw local-address?      inet:ip-address-no-zone
  |   |   +--rw remote-address?     inet:ip-address-no-zone
  |   |   +--rw routing-instance?   routing-instance-ref
  |   |   +--rw description?        string
  |   |   +--rw bind-interface?     if:interface-ref
  |   |   +--rw clear-df?            empty
  |   |   +--rw shutdown?           empty
  |   |   +--rw tmtu?                uint16
  |   |   +--rw mirror-destination? string
  |   |   +--rw hop-limit?          uint8
  |   |   +--rw tos?                 int8
  |   +--ro tunnel-state
  |   |   +--ro tunnels*
  |   |   |   +--ro name?            string
  |   |   |   +--ro type?            identityref
  |   |   |   +--ro local-ip?        inet:ip-address-no-zone
  |   |   |   +--ro remote-ip?       inet:ip-address-no-zone
  |   |   |   +--ro (state)?
  |   |   |   |   +--:(up)
  |   |   |   |   |   +--ro up?      empty
  |   |   |   |   +--:(down)
  |   |   |   |   |   +--ro down?    empty
  |   |   |   |   +--:(shutdown)
  |   |   |   |   |   +--ro shutdown? empty
  |   |   |   +--ro bind-interface? if:interface-state-ref
  |   |   |   +--ro user-configured? boolean
  |   |   |   +--ro routing-instance? routing-instance-ref
  |   |   |   +--ro tmtu?             uint16
  |   |   |   +--ro clear-df?         empty
  |   |   |   +--ro down-reason?      string
  |   |   |   +--ro resolved-interface-name? string
  |   |   |   +--ro hop-limit?        uint32
  |   |   |   +--ro tos?              int32
  |   |   +--ro tunnel-protocol?     identityref
  augment /if:interfaces-state/if:interface:
    +--ro tunnel-protocol? identityref

```

4. IP Tunnel YANG Data Model

```
<CODE BEGINS> file "ietf-ip-tunnel@2016-06-20.yang"
module ietf-ip-tunnel{

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

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-inet-types {
    prefix inet;
  }

  import ietf-routing {
    prefix "rt";
  }

  import iana-if-type {
    prefix ianaift;
  }

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group.";

  contact
    "Mandy.Liu@ericsson.com
    Adam.Foldes@ericsson.com
    zhengguangying@huawei.com";

  description
    "This YANG model defines the configuration data
    and operational state data for generic IPv4/6-in-IPv4 tunnel.
    It includes the IPv4 in IPv4, 6-to-4, and IPv6 over IPv4 manual
    tunnels.";

  revision 2016-04-27 {
    description
      "Made model more generic in order to allow augmentation by e.g.
      GRE tunnels.";
    reference
      "RFC XXXX: A YANG Data Model for IPv4 Tunnel.";
  }
  revision 2016-03-11 {
    description
```

```
    "Collapsed all tunnel types into a single subtree based on
      suggestions to more closely follow the IP Tunnel MIB.";
  reference
    "RFC XXXX: A YANG Data Model for IPv4 Tunnel.";
}
revision 2015-10-14 {
  description
    "Update model based on comments.";
  reference
    "RFC XXXX: A YANG Data Model for IPv4 Tunnel.";
}

revision 2015-07-20 {
  description
    "This version adds the following new items:
     - hop-limit
     - tos
     - tunnel-type
    This version changes 'ipv6to4-auto' to 'ipv6to4'";
  reference
    "RFC XXXX: A YANG Data Model for IPv4 Tunnel.";
}

revision 2015-05-27 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: A YANG Data Model for IPv4 Tunnel.";
}

/* Identities */
identity ip-tunnel-type {
  description
    "Base identity from which identities describing
     IP tunnel types are derived.";
}
identity ip-ip {
  base ip-tunnel-type;
  description
    "This identity represents IPv4-in-IPv4 tunnel type";
}
identity ipv6v4-manual {
  base ip-tunnel-type;
  description
    "This identity represents IPv6-to-IPv4 manual tunnel type";
}
identity ipv6-to-v4 {
  base ip-tunnel-type;
```

```
description
  "This identity represents the 6-to-4 tunnel type";
}

typedef routing-instance-ref {
  type leafref {
    path "/rt:routing/rt:routing-protocols/rt:routing-protocol/rt:name";
  }
  description
    "This type is used for leaves that reference a routing instance
    configuration.";
}

/*Configuration Data*/
container Tunnels{
  description
    "Configuration data for tunnels.";
  list Tunnel{
    key "name type";
    description
      "Configuration data for tunnels.";
    leaf name {
      type string;
      description
        "Name of the tunnel.";
    }
    leaf type {
      type identityref {
        base ip-tunnel-type;
      }
      description "The encapsulation type of the tunnel.";
    }

    leaf routing-instance {
      type routing-instance-ref;
      description "The routing instance of the local address.";
    }
    uses tunnel-config-components;
  }
}

/* Configuration data */
grouping tunnel-config-components {
  description
    "Configuration data for all tunnels and subtunnels";
  leaf description {
    type string {
      length "1..255";
    }
  }
}
```

```
    }
    description
      "Textual description for a tunnel. Can be any "+
      "alphanumeric string, including spaces, not to exceed "+
      "255 ASCII characters.";
  }
  leaf bind-interface {
    type if:interface-ref;
    description
      "Bind to an interface.";
  }
  leaf local-address {
    type inet:ip-address-no-zone;
    description "IP address of the local end of the tunnel.";
  }
  leaf remote-address {
    when "type != ipv6-to-v4" {
      description
        "6-to-4 tunnels do not have a fixed remote endpoint.";
    }
    type inet:ip-address-no-zone;
    description "IP address of the remote end of the tunnel.";
  }
}

leaf clear-df {
  type empty;
  description
    "If clear-df is absent, it means that fragmentation of
    tunnel packets are permitted. If clear-df is present,
    it means that fragmentation of tunnel packets are not
    permitted.";
}

leaf shutdown {
  type empty;
  description
    "Disable/enable the tunnel.";
}

leaf tmtu {
  type uint16 {
    range "256..16384";
  }
  description
    "Sets the Maximum Transmission Unit (MTU) size for
    packets sent in a tunnel. The default MTU is the MTU
    for the interface to which the tunnel is bound.";
}

leaf mirror-destination {
  type string;
}
```

```

    description
      "Designate the name of a tunnel as a circuit
       mirror destination. ";
  }
  leaf hop-limit {
    type uint8 {
      range "0|1..255";
    }
    description
      "The IPv4 TTL or IPv6 Hop Limit which is used in the
       outer IP header. A value of 0 indicates that the value
       is copied from the payload's header.";
  }
  leaf tos {
    type int8 {
      range "-1..63";
    }
    description
      "The method used to set the high 6 bits (the
       differentiated services codepoint) of the IPv4 TOS or
       IPv6 Traffic Class in the outer IP header. A value of -1
       indicates that the bits are copied from the payload's
       header. A value between 0 and 63 inclusive indicates
       that the bit field is set to the indicated value.";
  }
}

}

/*Operational state data*/
grouping tunnel-oper-states {
  description "Operational states of tunnels";
  choice state {
    description "Choice of operational states";
    case up {
      leaf up {
        type empty;
        description "The tunnel is up.";
      }
    }
    case down {
      leaf down {
        type empty;
        description "The tunnel is down.";
      }
    }
    case shutdown {
      leaf shutdown {
        type empty;
      }
    }
  }
}

```

```
        description "The tunnel is shut down administratively.";
    }
}
}
```

```
grouping tunnel-state-components {
  description
    "The basic tunnel information to be displayed.";

  leaf name {
    type string;
    description
      "Name of the tunnel.";
  }

  leaf type {
    type identityref;
    description
      "The type of the tunnel.";
  }

  leaf local-ip {
    type inet:ip-address-no-zone;
    description
      "IP address of the local end of the tunnel.";
  }

  leaf remote-ip {
    type inet:ip-address-no-zone;
    description
      "IP address of the remote end of the tunnel.";
  }

  uses tunnel-oper-states;
  leaf bind-interface {
    type if:interface-state-ref;
    description
      "Bind to an interface.";
  }

  leaf user-configured {
    type boolean;
    description
      "Indicate the tunnel is user-configured or dynamic.
      False is for dynamic.";
  }

  leaf routing-instance {
    type routing-instance-ref;
    description
      "Name of the reference routing instance. ";
  }
}
```

```
leaf tmtu {
  type uint16;
  description
    "The Maximum Transmission Unit (MTU) size for
    packets sent in a tunnel.";
}
leaf clear-df {
  type empty;
  description
    "Indicate that the DF bit is cleared.";
}
leaf down-reason {
  type string;
  description
    "The reason of the tunnel is down.";
}
leaf resolved-interface-name{
  type string;
  description
    "The egress interface name of the tunnel.";
}
leaf hop-limit {
  type uint32;
  description
    "The IPv4 TTL or IPv6 Hop Limit which is used in the outer IP
    header. A value of 0 indicates that the value is copied from
    the payload's header.";
}
leaf tos {
  type int32;
  description
    "The high 6 bits (the differentiated
    services codepoint) of the IPv4 TOS or IPv6 Traffic Class in
    the outer IP header. A value of -1 indicates that the bits
    are copied from the payload's header. A value between 0 and
    63 inclusive indicates that the bit field is set to the
    indicated value.";
}
}

container tunnel-state {
  config "false";
  description
    "Contain the information currently configured tunnels.";
  list tunnels {
    description
      "Operational state data of tunnels.";
    uses tunnel-state-components;
  }
}
```

```
    }  
  }  
  
  //Augment operational state data of IP interfaces  
  augment "/if:interfaces-state/if:interface" {  
    when "if:type = 'ianaift:tunnel'" {  
      description  
        "Augment IP interface.";  
    }  
    description  
      "Augment operational state data of IP interfaces.";  
    leaf tunnel-protocol {  
      type identityref;  
      description  
        "Indicate the state of the IP tunnel interface.";  
    }  
  }  
}
```

<CODE ENDS>

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 mandatory-to-implement secure transport is SSH [RFC6242] . The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

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-ip-tunnel

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

7. 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., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

Authors' Addresses

Ying Liu
Ericsson
No.5 Lize East Street
Beijing 100102
China

Email: Mandy.Liu@ericsson.com

Adam Mate Foldes
Ericsson
300 Holger Way
San Jose CA 95134
USA

Email: Adam.Foldes@ericsson.com

Guangying Zheng
Huawei Technologies, Co., Ltd
101 Software Avenue, Yuhua District
Nanjing 210012
China

Email: zhengguangying@huawei.com

Zitao Wang
Huawei Technologies, Co., Ltd
101 Software Avenue, Yuhua District
Nanjing 210012
China

Email: wangzitao@huawei.com

Yan Zhuang
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: zhuangyan.zhuang@huawei.com

Aijun Wang
China Telecom
No.118, Xizhimenneidajie, Xicheng District
Beijing 100035
China

Email: wangaj@ctbri.com.cn