

TRILL Working Group
INTERNET-DRAFT
Intended status: Standards Track

W. Hao
Y. Li
Huawei Technologies
D. Kumar
Cisco
M. Durrani
Cisco
H. Zhai
JIT
L. Xia
Huawei Technologies
December 21, 2015

Expires: June 2016

TRILL YANG Data Model
draft-ietf-trill-yang-04.txt

Abstract

This document defines a YANG data model for TRILL protocol.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

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
2.	Conventions Used in This Document.....	2
3.	Design of Data model.....	3
4.	TRILL YANG Data model.....	7
5.	Security Considerations.....	24
6.	IANA Considerations	25
7.	References	25
7.1.	Normative References.....	25
7.2.	Informative References.....	25
8.	Acknowledgments	25

[1.](#) Introduction

YANG [[RFC6020](#)] is a data definition language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [[RFC6241](#)].

This document defines a YANG [[RFC6020](#)] data model for the operation of TRILL base protocol.

[2.](#) Conventions Used in This Document

This document uses the acronyms defined in [[RFC6325](#)], in addition to the following:

CSNP: Complete Sequence Number Protocol Data Unit
DRB: Designated RBridge

IS-IS: Intermediate System to Intermediate System
LSDB: Link State Database
MAC: Media Access Control address
MTU: Maximum Transmission Unit
NETCONF: Network Configuration Protocol
PSNP: Partial Sequence Number Packet
RBridge: An alternative name for a TRILL Switch
RPF: Reverse Path Forward
SNP: Scalable Network Pack
SSH: Secure Shell
VLAN: Virtual Local-Area Network

3. Design of Data model

There is only one module for the TRILL base protocol. The module can be augmented for other TRILL extended features with their specific definitions, such as TRILL active-active connection, TRILL Fine Grained Label, etc.

The TRILL Yang module includes one container of trillSites which contains a list of instances as many implementations are currently supporting multiple ISIS instances within a single RBridge. The configuration data is divided into four categories which include per RBridge, per nickname per RBridge, per port per RBridge, and per VLAN per RBridge. The operating status includes the information of LSDB, unicast and multicast routing table, RPF check, nickname, peer and some statistics.

The figure below describes the overall structure of the TRILL Yang model:

```

module: ietf-trill
  +-rw trill-site
    +-rw max-loadbalance-num?      uint8
    +-rw bandwidth-reference?     uint32
    +-rw native-confidence?       uint8
    +-rw remote-confidence?       uint8
    +-rw min-link-mtu?            uint16
    +-rw mtu-probes?              uint8
    +-rw lsp-timer
      | +-rw lspLife?              uint16
      | +-rw lsp-refresh?         uint16
      | +-rw lsp-generation?     uint8
    +-rw lsp-snp-authen
      | +-rw authen-mode?         authen-mode
      | +-rw cipher-or-plain?     boolean
      | +-rw password-key?       string
      | +-rw password?           string
      | +-rw keychain?           string
    +-rw tree-para
      | +-rw nick-number?        uint16
      | +-rw tree-number?        uint16
    +-rw net-entity*              string
    +-rw nicknames
      | +-rw nickname* [nickName]
      |   +-rw nickName          uint16
      |   +-rw priority?         uint8
      |   +-rw root-priority?    uint16
    +-rw trill-ports
      | +-rw trill-port* [ifName]
      |   +-rw ifName              string
      |   +-rw port-mode?          port-mode
      |   +-rw hello-authen
      |     | +-rw authen-mode?    authen-mode
      |     | +-rw cipher-or-plain? boolean
      |     | +-rw password-key?   string
      |     | +-rw password?       string
      |     | +-rw keychain?       string
      |   +-rw csnp-timer?         uint16
      |   +-rw hello?             uint8
      |   +-rw holding-multiplier? uint16
      |   +-rw lsp-retransmit?     uint16
      |   +-rw lsp-throttle
      |     | +-rw throttleInterval? uint16
      |     | +-rw count-number?     uint16
      |   +-rw inhibition-timer?   uint8
      |   +-rw drb-config

```

```

|   |   +--rw drb-priority?    uint8
|   |   +--rw holding-timer?  uint8
|   +--rw mac-learning-flag?   boolean
|   +--rw trill-frame-receive-flag? boolean
|   +--rw cost?                uint32
|   +--rw enabled-vlans?       binary
|   +--rw announcing-vlans?    binary
|   +--rw forwarding-vlans?    binary
|   +--rw designated-vlan?     uint16
+--rw vlan-param
|   +--rw vlan-config* [vlan-id]
|       +--rw vlan-id          uint16
|       +--rw participation-flag? boolean
|       +--rw priority?        uint8
|       +--rw holdingTimer?    uint8
+--rw trill-route-infos
|   +--ro trill-route-info* [nickname nexthop]
|       +--ro nickname          uint32
|       +--ro cost?             uint32
|       +--ro out-interface?    string
|       +--ro out-vlan?         uint32
|       +--ro nexthop           string
|       +--ro hop-count?        uint32
+--rw trill-mroute-infos
|   +--ro trill-mroute-info* [vlan root-nickname]
|       +--ro vlan              uint16
|       +--ro root-nickname     uint16
|       +--ro hop-count?        uint16
|       +--ro trill-mroute-outinterface-info*
|           [out-interface out-vlan]
|           +--ro out-interface  string
|           +--ro out-vlan       uint32
+--rw trill-rpf-check-infos
|   +--ro trill-rpf-check-info* [ingress-nickname tree-nickname]
|       +--ro ingress-nickname  uint16
|       +--ro tree-nickname     uint16
|       +--ro interface-name?   string
|       +--ro neighbor-mac?     string
|       +--ro out-vlan?         uint16
+--rw trill-peer-infos
|   +--ro trill-peer-info* [host-name circuit-id]
|       +--ro host-name         string
|       +--ro interface-name?   string
|       +--ro circuit-id        string
|       +--ro status?           peer-state
|       +--ro holdTime?         uint32
|       +--ro priority?         string

```

```
+--rw trill-lsdb-infos
| +--ro trill-lsdb-info* [lspid]
|   +--ro lspid          string
|   +--ro sequence-number? string
|   +--ro checksum?      string
|   +--ro lsp-length?    uint32
|   +--ro lsp-flags?     bits
|   +--ro holdtime?      string
|   +--ro local-lsp?     boolean
+--rw trill-nickname-infos
| +--ro trill-nickname-info* [nickname systemid]
|   +--ro nickname       uint32
|   +--ro priority?      uint32
|   +--ro root-priority? uint32
|   +--ro systemid       string
|   +--ro conflict-state? conflict-state
|   +--ro static-flag?   static-flag
|   +--ro is-local?      boolean
+--ro trill-statistics
| +--ro interface-stat
| | +--ro up-num?      uint32
| | +--ro down-num?   uint32
| +--ro pktstatistics
| | +--ro report-num?  uint32
| | +--ro detect-num?  uint32
| | +--ro twoway-num?  uint32
| +--ro unicast-routes-num? uint32
| +--ro multicast-routes-num? uint32
| +--ro rpf-entrys-num?    uint32
| +--ro remote-nicknames-num? uint32
| +--ro lsdb-lsps-num?     uint32
| +--ro self-lsps-num?     uint32
| +--ro multicast-trees-num? uint32
| +--ro unicast-nodes-num?  uint32
| +--ro multicast-nodes-num? uint32
+--rw pkt-statistics
  +--ro pkt-statistic* [interface-name]
    +--ro interface-name      string
    +--ro sent-hellos-num?    uint32
    +--ro recved-hellos-num?  uint32
    +--ro sent-lsps-num?      uint32
    +--ro recved-lsps-num?    uint32
    +--ro sent-csnps-num?     uint32
    +--ro recved-csnps-num?   uint32
    +--ro sent-psnps-num?     uint32
    +--ro recved-psnps-num?   uint32
    +--ro lsp-retransmissions-num? uint32
```

+--ro drb-elections-num? uint32

4. TRILL YANG Data model

<CODE BEGINS> file "ietf-trill@2015-12-21.yang"

```
module ietf-trill {
  yang-version 1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-trill";
  //namespace need to be assigned by IANA
  prefix trill;

  organization "IETF TRILL Working Group";
  contact
    "WG Web: <http://tools.ietf.org/wg/TRILL/>
    WG List: <mailto:trill@ietf.org>

  WG Chair: Susan Hares
             <mailto:shares@ndzh.com>

  Editor: Weiguo Hao
          <mailto:haoweiguo@huawei.com>;

  description "TRILL Yang Module";
  revision 2015-12-21{
    description "04 version, as per RFC6325";
    reference "draft-ietf-trill-yang-04";
  }
  typedef port-mode {
    type enumeration {
      enum access {
        value "0";
        description "Access Port";
      }
      enum p2p {
        value "1";
        description "P2P Hello Link";
      }
      enum trunk {
        value "2";
        description "Trunk Port";
      }
      enum hybrid {
        value "3";
        description "Both Trunk and Access Port";
      }
    }
  }
```

```
    }
  }
  description "TRILL port mode type";
}
typedef drb-state {
  type enumeration {
    enum non-drb {
      value "0";
      description "Non-DRB Port";
    }
    enum drb {
      value "1";
      description "DRB Port";
    }
    enum down {
      value "2";
      description "Down Port";
    }
    enum suspended {
      value "3";
      description "Suspended Port";
    }
  }
}
description "TRILL DRB port state type.";
}
typedef peer-state {
  type enumeration {
    enum report {
      value "0";
      description "report state";
    }
    enum detect {
      value "1";
      description "detect state";
    }
    enum down {
      value "2";
      description "down state";
    }
    enum 2way {
      value "3";
      description "2way state";
    }
  }
}
description "TRILL neighbour state type.";
}
typedef authen-mode {
```



```
type enumeration {
  enum "simple" {
    value "0";
    description "Indicates that the password is transmitted
in plaintext";
  }
  enum "md5" {
    value "1";
    description "Indicates that the password is transmitted
after encryption by MD5.";
  }
  enum "keychain" {
    value "2";
    description "Indicates that the password is a keychain
which varies with time.";
  }
}
description "TRILL authentication mode.";
}
grouping authen-para {
  leaf authen-mode {
    type authen-mode;
    description "authentication-mode ";
  }
  leaf cipher-or-plain {
    type boolean;
    description "0 indicates that the password is in ciphertext,
1 indicates that the password is in plaintext";
  }
  leaf password-key {
    type string;
    description "The value is a string of case-sensitive
characters, spaces not supported. The password range varies
with the authentication mode. When the authentication mode
is simple, the value is a string of 1 to 16 characters
in simple text, or a string of 32 characters in ciphertext.
When the authentication mode is md5, the value is a string
of 1 to 255 characters in simple text, or a string of 32 to
392 characters in ciphertext.";
  }
  leaf password {
    type string;
    description "The value is a string of case-sensitive
characters,
spaces not supported. The password range varies with the
authentication mode. When the authentication mode is simple,
```

```
    the password ranges from 1 to 16 characters. When the
    authentication mode is md5, the password ranges
    from 1 to 255 characters.";
  }
  leaf keychain {
    type string;
    description "The name is keychain name which is a string
    of 1 to 47 case-insensitive characters, spaces not supported.";
  }
  description "TRILL protocol authentication configurations.";
}
typedef conflict-state {
  type enumeration {
    enum "S" {
      value "0";
      description "Nickname suspended";
    }
    enum "A" {
      value "1";
      description "Nickname advertised";
    }
  }
  description "Nickname conflict state.";
}
typedef static-flag{
  type enumeration {
    enum "S" {
      value "0";
      description "Manually configured";
    }
    enum "D" {
      value "1";
      description "Dynamic generated";
    }
  }
  description "Static or dynamic nickname flag.";
}

container trill-site {
  description "Trill all configuration and display information";
  leaf max-loadbalance-num {
    type uint8 {
      range "1..max";
    }
    default "32";
    description "The maximum number of equal-cost
    routes for load-balancing.";
```

```
}
leaf bandwidth-reference {
  type uint32 {
    range "1..2147483648";
  }
  default "200000000";
  description "Sets the bandwidth reference value
for a TRILL interface. Unit:M bit/s.
Cost of the interface = Bandwidth reference
value/Interface bandwidth";
}
leaf native-confidence {
  type uint8;
  default "32";
  description "The confidence in { MAC, VLAN, local port }
triples learned from locally received native frames";
}
leaf remote-confidence {
  type uint8;
  default "32";
  description "The confidence in { MAC, VLAN, remote RBridge}
triples learned from decapsulating frames";
}
leaf min-link-mtu {
  type uint16 {
    range "1..max";
  }
  default "1470";
  description "Minimum link MTU";
}
leaf mtu-probes {
  type uint8 {
    range "1..max";
  }
  default "3";
  description "The number of failed MTU-probes";
}
container lsp-timer {
  leaf lspLife {
    type uint16 {
      range "2..max";
    }
    default "1200";
    description "LSP aging timer.Unit:Second";
  }
  leaf lsp-refresh {
    type uint16 {
```

```
        range "1..65534";
    }
    default "900";
    description "LSP refresh timer.Unit:Second";
}
leaf lsp-generation {
    type uint8 {
        range "1..120";
    }
    default "2";
    description "LSP generate timer.Unit:Second";
}
description "LSP timer configuration";
}
container lsp-snp-authen {
    uses authen-para;
    description "Authentication information for LSPs and SNPs";
}
container tree-para {
    leaf nick-number {
        type uint16 {
            range "1..256";
        }
        default "1";
        description "Nickname number";
    }
    leaf tree-number {
        type uint16 {
            range "1..max";
        }
        default "1";
        description "Distribution tree number";
    }
    description "Distribution tree configuration";
}

leaf-list net-entity {
    type string;
    description "TRILL network entity";
}

container nicknames {
    description "All Nicknames and the associated properties";
    list nickname {
        key "nickName";
        description "Each nickname and the associated property";
        leaf nickName {
```

```
        type uint16 {
            range "1..65471";
        }
        description "Nickname value";
    }
    leaf priority {
        type uint8 {
            range "128..max";
        }
        default "192";
        description "Nickname priority";
    }
    leaf root-priority {
        type uint16 {
            range "1..max";
        }
        default "32768";
        description "Nickname root priority";
    }
}
}
container trill-ports {
    description "All TRILL port configurations";
    list trill-port {
        key "ifName";
        description "Each TRILL port configuration";
        leaf ifName {
            type string;
            description "trill interface";
        }
        leaf port-mode {
            type port-mode;
            default "p2p";
            description "TRILL port mode";
        }
        container hello-authen {
            uses authen-para;
            description "Authentication information for Hellos";
        }
        leaf csnp-timer {
            type uint16 {
                range "1..max";
            }
            default "10";
            description "Csnp-timer value";
        }
        leaf hello {
```

```
    type uint8 {
      range "3..max";
    }
    default "10";
    description "Hello timer value";
  }
  leaf holding-multiplier {
    type uint16 {
      range "3..1000";
    }
    default "3";
    description "Holding-multiplier timer value";
  }
  leaf lsp-retransmit {
    type uint16 {
      range "1..300";
    }
    default "5";
    description "Lsp retransmit timers";
  }
  container lsp-throttle {
    description "TRILL LSP throttle configuration.";
    leaf throttleInterval {
      type uint16 {
        range "1..10000";
      }
      default "50";
      description "The interval timer between
        two LSP messages.Unit:ms";
    }
    leaf count-number {
      type uint16 {
        range "1..1000";
      }
      default "10";
      description "The max messages number being
        sent each time.Unit:ms";
    }
  }
  leaf inhibition-timer {
    type uint8 {
      range "0..30";
    }
    default "30";
    description "The inhibition time for the port
      when root bridge changes.Unit:Second";
  }
}
```

```
container drb-config {
  description "TRILL DRB negotiation parameter";
  leaf drb-priority {
    type uint8 {
      range "0..127";
    }
    default "64";
    description "Drb priority";
  }
  leaf holding-timer {
    type uint8 {
      range "3..255";
    }
    default "10";
    description "Holding timer";
  }
}
leaf mac-learning-flag {
  type boolean;
  default "true";
  description "if learning MAC address from locally
  received native frames";
}
leaf trill-frame-receive-flag {
  type boolean;
  default "false";
  description "if receiving of TRILL frames from
  non IS-IS adjacency";
}
leaf cost {
  type uint32 {
    range "0..16777215";
  }
  default "0";
  description "link cost";
}
leaf enabled-vlans {
  type binary{
    length "1..512";
  }
  description "Enabled vlans";
}
leaf announcing-vlans {
  type binary{
    length "1..512";
  }
}
```

```
        description "Announcing vlans";
    }
    leaf forwarding-vlans {
        type binary{
            length "1..512";
        }
        description "Forwarding vlans";
    }
    leaf designated-vlan {
        type uint16{
            range "1..4096";
        }
        description "Designated vlans";
    }
}

container vlan-para {
    description "All VLAN parameters";
    list vlan-config{
        key vlan-id;
        description "Each VLAN parameter";
        leaf vlan-id {
            type uint16 {
                range "1..4096";
            }
        }
        description "Vlan Id";
    }
    leaf participation-flag {
        type boolean;
        default "false";
        description "Participation flag";
    }
    leaf priority {
        type uint8 {
            range "0..127";
        }
        default "64";
        description "Vlan priority";
    }
    leaf holdingTimer {
        type uint8 {
            range "3..max";
        }
        default "10";
        description "Holding timer";
    }
}
```



```
    }
  }

  container trill-route-infos {
    description "All TRILL unicast route informations";
    list trill-route-info {
      key "nickname nexthop";
      config "false";
      description "Each TRILL unicast route information";

      leaf nickname {
        type uint32;
        config "false";
        description "Dest nickname for the route";
      }
      leaf cost {
        type uint32;
        config "false";
        description "The link cost to the nickname";
      }
      leaf out-interface {
        type string;
        config "false";
        description "Out interface to the nickname";
      }
      leaf out-vlan {
        type uint32;
        config "false";
        description "Out vlan on the out interface";
      }
      leaf nexthop {
        type string;
        config "false";
        description "Next hop to the nickname";
      }
      leaf hop-count {
        type uint32;
        config "false";
        description "Hop count to the nickname";
      }
    }
  }

  container trill-mroute-infos {
    list trill-mroute-info {
      key "vlan root-nickname";
      config "false";
```

```
    description "Distribution pruning tree route table;
    For non-pruning tree, VLAN is set to be 0XFFFF";

    leaf vlan {
      type uint16;
    }
    description "CE vlan";
  }
  leaf root-nickname {
    type uint16;
    description "Root nickname";
  }
  leaf hop-count {
    type uint16;
    description "Hop count";
  }
  list trill-mroute-outinterface-info {
    key "out-interface out-vlan";
    description "The out interface and out vlan for a multicast
route";

    leaf out-interface {
      type "string";
      description "Out interface";
    }
    leaf out-vlan {
      type "uint32";
      description "Out vlan";
    }
  }
}
description "TRILL multicast route information";
}

container trill-rpf-check-infos {
  list trill-rpf-check-info {
    key "ingress-nickname tree-nickname";
    config "false";
    description "Each TRILL RPF check information";

    leaf ingress-nickname {
      type "uint16";
      description "Ingress nickname";
    }
    leaf tree-nickname {
      type "uint16";
      description "Tree nickname";
    }
  }
}
```

```
    leaf interface-name {
      type "string";
      description "Interface name";
    }
    leaf neighbor-mac {
      type "string";
      description "Neighbour Mac";
    }
    leaf out-vlan {
      type "uint16";
      description "Out vlan";
    }
  }
  description "All TRILL RPF check information";
}

container trill-peer-infos {
  list trill-peer-info {
    key "host-name circuit-id";
    config "false";
    description "Each TRILL peer information";

    leaf host-name {
      type "string";
      description "Peer RBridge name";
    }
    leaf interface-name {
      type "string";
      description "Interface name";
    }
    leaf circuit-id {
      type "string";
      description "Circuit Id";
    }
    leaf status {
      type peer-state;
      description "Peer status";
    }
    leaf holdTime {
      type "uint32";
      description "Hold time";
    }
    leaf priority {
      type "string";
      description "Priority";
    }
  }
}
```

```
    description "All TRILL peer informations";
  }

  container trill-lsdb-infos{
    description "All TRILL LSDB information";
    list trill-lsdb-info{
      key "lspid";
      config "false";
      description "Each TRILL LSDB information";

      leaf lspid {
        type string;
        description "Lspid";
      }
      leaf sequence-number {
        type string;
        description "Sequence number";
      }
      leaf checksum {
        type string;
        description "Checksum";
      }
      leaf lsp-length {
        type uint32 {
          range "0..2000";
        }
        description "Lsp length";
      }
    }

    leaf lsp-flags {
      type bits {
        bit attbit {
          position 0;
          description "TRILL lsp attribute flag";
        }
        bit partition-bit {
          position 1;
          description "TRILL lsp partition flag";
        }
        bit overload-bit {
          position 2;
          description "TRILL lsp position flag";
        }
      }
      default "attbit";
      description "TRILL lsp flags";
    }
  }
```

```
    leaf holdtime {
      type string;
      description "Hold time";
    }
    leaf local-lsp {
      type boolean;
      description "Local lsp";
    }
  }
}

container trill-nickname-infos {
  description "All Trill nickname information";
  list trill-nickname-info {
    key "nickname systemid";
    config "false";
    description "Each Trill nickname information";

    leaf nickname {
      type uint32;
      description "Nickname";
    }
    leaf priority {
      type uint32;
      description "Priority";
    }
    leaf root-priority {
      type uint32;
      description "Root priority";
    }
    leaf systemid {
      type string;
      description "SystemId";
    }
    leaf conflict-state {
      type conflict-state;
      description "Conflict state";
    }
    leaf static-flag {
      type static-flag;
      description "Static flag";
    }
    leaf is-local {
      type boolean;
      description "Is local flag";
    }
  }
}
```

```
    }  
  }  
  container trill-statistics {  
    config "false";  
    description "Trill statistics";  
  
    container interface-stat {  
      leaf up-num {  
        type uint32;  
        description "Trill up interface number";  
      }  
      leaf down-num {  
        type uint32;  
        description "Trill down interface number";  
      }  
      description "Trill up and down interface number";  
    }  
    container pktstatistics {  
      leaf report-num {  
        type uint32;  
        description "Trill interface number in report state";  
      }  
      leaf detect-num {  
        type uint32;  
        description "Trill interface number in detect state";  
      }  
      leaf twoway-num {  
        type uint32;  
        description "Trill interface number in two way state";  
      }  
      description "Trill packet statistics";  
    }  
    leaf unicast-routes-num {  
      type uint32;  
      description "Trill unicast route number";  
    }  
    leaf multicast-routes-num {  
      type uint32;  
      description "Trill multicast route number";  
    }  
    leaf rpf-entrys-num {  
      type uint32;  
      description "Trill RPF entry number";  
    }  
    leaf remote-nicknames-num {  
      type uint32;  
      description "Trill remote nickname number";  
    }  
  }  
}
```

```
    }
    leaf lsdb-lsps-num {
        type uint32;
        description "Trill LSP number";
    }
    leaf self-lsps-num {
        type uint32;
        description "Trill self generated LSP number";
    }
    leaf multicast-trees-num {
        type uint32;
        description "Trill distribution tree tree number";
    }
    leaf unicast-nodes-num {
        type uint32;
        description "Trill unicast node number";
    }
    leaf multicast-nodes-num {
        type uint32;
        description "Trill multicast node number";
    }
}

container pkt-statistics {
    list pkt-statistic {
        key "interface-name";
        config "false";

        description "Each packet statistic";
        leaf interface-name {
            type string;
            description "Interface name";
        }
        leaf sent-hellos-num {
            type uint32;
            description "The number of the sent TRILL hello";
        }
        leaf recved-hellos-num {
            type uint32;
            description "The number of the received TRILL hello";
        }
        leaf sent-lsps-num {
            type uint32;
            description "The number of the sent LSP number";
        }
        leaf recved-lsps-num {
            type uint32;
```

```
        description "The number of the received TRILL LSP";
    }
    leaf sent-csnps-num {
        type uint32;
        description "The number of the sent CSNP number";
    }
    leaf recved-csnps-num {
        type uint32;
        description "The number of the received CSNP number";
    }
    leaf sent-psnps-num {
        type uint32;
        description "The number of the sent PSNP number";
    }
    leaf recved-psnps-num {
        type uint32;
        description "The number of the received PSNP number";
    }
    leaf lsp-retransmissions-num {
        type uint32;
        description "The number of the retransmitted LSP number";
    }
    leaf drb-elections-num {
        type uint32;
        description "The number of the DRB election number";
    }
}
description "All packet statistics";
}
```

<CODE ENDS>

5. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241] [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242] [RFC6242]. The NETCONF access control model [RFC6536] [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 requires no IANA Actions. RFC Editor: Please remove this section before publication.

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", [RFC 2234](#), Internet Mail Consortium and Demon Internet Ltd., November 1997.

7.2. Informative References

- [RFC6325] Perlman, R., et.al., "Routing Bridges (RBridges): Base Protocol Specification", [RFC 6325](#), July 2011.
- [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)", [RFC6241](#), June 2011.

8. Acknowledgments

The authors wish to acknowledge the important contributions of Donald Eastlake, Susan Hares, Guangying Zheng, Xianping Zhang, Wenxia Hou, Zhibo Hu.

Authors' Addresses

Weiguo Hao
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China
Phone: +86-25-56623144
Email: haoweiguo@huawei.com

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China
Phone: +86-25-56625375
Email: liyizhou@huawei.com

Deepak Kumar
CISCO Systems
510 McCarthy Blvd
Milpitas, CA 95035.
Email: dekumar@cisco.com

Muhammad Durrani
Cisco
Email: mdurrani@cisco.com

Hongjun Zhai
Jinling Institute of Technology
99 Hongjing Avenue, Jiangning District
Nanjing, Jiangsu 211169
China
Email: hongjun.zhai@tom.com

Liang Xia
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China
Email: frank.xialiang@huawei.com