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/1id-abstracts.html](http://www.ietf.org/1id-abstracts.html)

The list of Internet-Draft Shadow Directories can be accessed at [http://www.ietf.org/shadow.html](http://www.ietf.org/shadow.html).
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
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-para
|     |  +--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-entries-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
module ietf-trill {
    yang-version 1;
    namespace "urn:ietf:params:xml:ns:yang:trill";
    //namespace need to be assigned by IANA
    prefix trill;

    organization "IETF TRILL Working Group";
    description "TRILL Yang Module";
    revision 2015-08-06{
        description "03 version, as per RFC6325";
        reference "draft-ietf-trill-yang-03";
    }
    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.";
    }
    typedef drb-state {
        type enumeration {
            enum non-drb {
                value "0";
                description "Non-DRB Port";
            }
            enum drb {

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.";
}
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.";
}

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";
    }
}
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 {
    leaf max-loadbalance-num {
        default "32";
        type uint8 {
            range "1..max";
        }
        description "The maximum number of equal-cost routes for load-balancing.";
    }
    leaf bandwidth-reference {
        default "20000000";
        type uint32 {
            range "1..2147483648";
        }
        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 {
        default "32";
        type uint8;
    }
}
description "The confidence in { MAC, VLAN, local port } triples learned from locally received native frames";
}
leaf remote-confidence {
  default "32";
  type uint8;
  description "The confidence in { MAC, VLAN, remote RBridge} triples learned from decapsulating frames";
}
leaf min-link-mtu {
  default "1470";
  type uint16 {
    range "1..max";
  }
  description "Minimum link MTU";
}
leaf mtu-probes {
  default "3";
  type uint8 {
    range "1..max";
  }
  description "The number of failed MTU-probes";
}
container lsp-timer {
  leaf lspLife {
    default "1200";
    type uint16 {
      range "2..max";
    }
    description "LSP aging timer.Unit:Second";
  }
  leaf lsp-refresh {
    default "900";
    type uint16 {
      range "1..65534";
    }
    description "LSP refresh timer.Unit:Second";
  }
  leaf lsp-generation {
    default "2";
    type uint8 {
      range "1..120";
    }
    description "LSP generate timer.Unit:Second";
  }
}
container lsp-snp-authen {
uses authen-para;
description "Authentication information for LSPs and SNPs";
}

container tree-para {
  leaf nick-number {
    default "1";
    type uint16 {
      range "1..256";
    }
    description "Nickname number";
  }
  leaf tree-number {
    default "1";
    type uint16 {
      range "1..max";
    }
    description "Distribution tree number";
  }
}

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

container nicknames {
  list nickname {
    key "nickName";
    leaf nickName {
      type uint16 {
        range "1..65535";
      }
    }
    leaf priority {
      default "192";
      type uint8 {
        range "128..max";
      }
    }
    leaf root-priority {
      default "32768";
      type uint16 {
        range "1..max";
      }
    }
  }
  description "Nicknames and the associated properties";
container trill-ports {
  list trill-port{
    key "ifName";
    leaf ifName {
      type string;
      description "trill interface";
    }
    leaf port-mode {
      default "p2p";
      type port-mode;
      description "TRILL Port Mode";
    }
  }
  container hello-authen {
    uses authen-para;
    description "Authentication information for Hellos";
  }
  leaf csnp-timer {
    default "10";
    type uint16 {
      range "1..max";
    }
  }
  leaf hello {
    default "10";
    type uint8 {
      range "3..max";
    }
  }
  leaf holding-multiplier {
    default "3";
    type uint16 {
      range "3..1000";
    }
  }
  leaf lsp-retransmit {
    default "5";
    type uint16 {
      range "1..300";
    }
  }
  container lsp-throttle {
    leaf throttleInterval {
      default "50";
      type uint16 {
        range "1..10000";
      }
    }
  }
}
description "The interval timer between two LSP messages. Unit: ms";
}
leaf count-number {
  default "10";
  type uint16 {
    range "1..1000";
  }
  description "The max messages number being sent each time. Unit: ms";
}
leaf inhibition-timer {
  default "30";
  type uint8 {
    range "0..30";
  }
  description "The inhibition time for the port when root bridge changes. Unit: Second";
}

container drb-config {
  leaf drb-priority {
    default "64";
    type uint8 {
      range "0..127";
    }
  }
  leaf holding-timer {
    default "10";
    type uint8 {
      range "3..255";
    }
  }
}
leaf mac-learning-flag {
  default "true";
  type boolean;
  description "if learning MAC address from locally received native frames";
}
leaf trill-frame-receive-flag {
  default "false";
  type boolean;
  description "if receiving of TRILL frames from non IS-IS adjacency";
}
leaf cost {
    default "0";
    type uint32 {
        range "0..16777215";
    }
    description "link cost";
}

leaf enabled-vlans {
    type binary {
        length "1..512";
    }
}

leaf announcing-vlans {
    type binary {
        length "1..512";
    }
}

leaf forwarding-vlans {
    type binary {
        length "1..512";
    }
}

leaf designated-vlan {
    type uint16 {
        range "1..4096";
    }
}

}

container vlan-para {
    list vlan-config {
        key vlan-id;

        leaf vlan-id {
            type uint16 {
                range "1..4096";
            }
        }

        leaf participation-flag {
            default "false";
            type boolean;
        }

        leaf priority {
            default "64";
            type uint8 {
                range "0..127";
            }
        }
    }
}

leaf holdingTimer {
    default "10";
    type uint8 {
        range "3..max";
    }
}

container trill-route-infos {
    list trill-route-info {
        key "nickname nexthop";
        config "false";

        leaf nickname {
            config "false";
            type uint32;
        }
        leaf cost {
            config "false";
            type uint32;
        }
        leaf out-interface {
            config "false";
            type string;
        }
        leaf out-vlan {
            config "false";
            type uint32;
        }
        leaf nexthop {
            config "false";
            type string;
        }
        leaf hop-count {
            config "false";
            type uint32;
        }
    }
}

container trill-mroute-infos {
    list trill-mroute-info {
        description "Distribution pruning tree route table;
For non-pruning tree, VLAN is set to be 0XFFFF";

key "vlan root-nickname";
config "false";

leaf vlan {
  type uint16;
}
leaf root-nickname {
  type uint16;
}
leaf hop-count {
  type uint16;
}
list trill-mroute-outinterface-info {
  key "out-interface out-vlan";
  leaf out-interface {
    type "string";
  }
  leaf out-vlan {
    type "uint32";
  }
}
}

container trill-rpf-check-infos {
  list trill-rpf-check-info {
    key "ingress-nickname tree-nickname";
    config "false";
    leaf ingress-nickname {
      type "uint16";
    }
    leaf tree-nickname {
      type "uint16";
    }
    leaf interface-name {
      type "string";
    }
    leaf neighbor-mac {
      type "string";
    }
    leaf out-vlan {
      type "uint16";
    }
  }
}

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

        leaf host-name {
            type "string";
            description "Peer RBridge name";
        }

        leaf interface-name {
            type "string";
        }

        leaf circuit-id {
            type "string";
        }

        leaf status {
            type peer-state;
        }

        leaf holdTime {
            type "uint32";
        }

        leaf priority {
            type "string";
        }
    }
}

container trill-lsdb-infos{
    list trill-lsdb-info{
        key "lspid";
        config "false";

        leaf lspid {
            type string;
        }

        leaf sequence-number {
            type string;
        }

        leaf checksum {
            type string;
        }

        leaf lsp-length {
            type uint32 {
                range "0..2000";
            }
        }
    }
}
leaf lsp-flags {
    type bits {
        bit attbit {
            position 0;
        }
        bit partition-bit {
            position 1;
        }
        bit overload-bit {
            position 2;
        }
    }
    default "attbit";
}

leaf holdtime {
    type string;
}

leaf local-lsp {
    type boolean;
}
}

container trill-nickname-infos {
    list trill-nickname-info {
        key "nickname systemid";
        config "false";

        leaf nickname {
            type uint32;
        }
        leaf priority {
            type uint32;
        }
        leaf root-priority {
            type uint32;
        }
        leaf systemid {
            type string;
        }
        leaf conflict-state {
            type conflict-state;
        }
        leaf static-flag {

type static-flag;
}
leaf is-local {
  type boolean;
}
}

container trill-statistics {
  config "false";
  container interface-stat {
    leaf up-num {
      type uint32;
    }
    leaf down-num {
      type uint32;
    }
  }
  container pktstatistics {
    leaf report-num {
      type uint32;
    }
    leaf detect-num {
      type uint32;
    }
    leaf twoway-num {
      type uint32;
    }
    leaf unicast-routes-num {
      type uint32;
    }
    leaf multicast-routes-num {
      type uint32;
    }
    leaf rpf-entrys-num {
      type uint32;
    }
    leaf remote-nicknames-num {
      type uint32;
    }
    leaf lsdb-lsps-num {
      type uint32;
    }
    leaf self-lsps-num {
      type uint32;
    }
    leaf multicast-trees-num {
type uint32;
}
leaf unicast-nodes-num {
    type uint32;
}
leaf multicast-nodes-num {
    type uint32;
}
}

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

        leaf interface-name {
            type string;
        }
        leaf sent-hellos-num {
            type uint32;
        }
        leaf recved-hellos-num {
            type uint32;
        }
        leaf sent-lsps-num {
            type uint32;
        }
        leaf recved-lsps-num {
            type uint32;
        }
        leaf sent-csnps-num {
            type uint32;
        }
        leaf recved-csnps-num {
            type uint32;
        }
        leaf sent-psnps-num {
            type uint32;
        }
        leaf recved-psnps-num {
            type uint32;
        }
        leaf lsp-retransmissions-num {
            type uint32;
        }
        leaf drb-elections-num {
            type uint32;
        }
    }
}
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


7.2. Informative References


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