Internet Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 2018

Y. Jiang, Ed.
Huawei
X. Liu
Independent
J. Xu
Huawei
R. Cummings, Ed.
National Instruments

November 28, 2017

YANG Data Model for IEEE 1588-2008
draft-ietf-tictoc-1588v2-yang-07

Abstract

This document defines a YANG data model for the configuration of IEEE 1588-2008 devices and clocks, and also retrieval of the configuration information, data set and running states of IEEE 1588-2008 clocks.

Table of Contents

1. Introduction

As a synchronization protocol, IEEE 1588-2008 [IEEE1588] is widely supported in the carrier networks, industrial networks, automotive networks, and many other applications. It can provide high precision time synchronization as fine as nano-seconds. The protocol depends on a Precision Time Protocol (PTP) engine to decide its own state automatically, and a PTP transportation layer to carry the PTP timing and various quality messages. The configuration parameters and state data sets of IEEE 1588-2008 are numerous.

According to the concepts described in [RFC3444], IEEE 1588-2008 itself provides an information model in its normative specifications for the data sets (in IEEE 1588-2008 clause 8). Some standardization organizations including the IETF have specified data models in MIBs (Management Information Bases) for IEEE 1588-2008 data sets (e.g. [RFC8173], [IEEE8021AS]). These MIBs are typically focused on retrieval of state data using the Simple Network Management Protocol (SNMP), furthermore, configuration of PTP data sets is not considered in [RFC8173].

Some service providers and applications require that the management of the IEEE 1588-2008 synchronization network be flexible and more Internet-based (typically overlaid on their transport networks). Software Defined Network (SDN) is another driving factor, which demands an improved configuration capability of synchronization networks.

YANG [RFC6020] is a data modeling language used to model configuration and state data manipulated by network management protocols like the Network Configuration Protocol (NETCONF) [RFC6241]. A small set of built-in data types are defined in [RFC6020], and a collection of common data types are further defined in [RFC6991]. Advantages of YANG include Internet based configuration capability, validation, rollback and so on. All of these characteristics make it attractive to become another candidate modeling language for IEEE 1588-2008.

This document defines a YANG [RFC6020] data model for the configuration of IEEE 1588-2008 devices and clocks, and retrieval of the state data of IEEE 1588-2008 clocks. The data model is based on the PTP data sets as specified in [IEEE1588]. The technology specific IEEE 1588-2008 information, e.g., those specifically implemented by a bridge, a router or a telecom profile, is out of scope of this document.

When used in practice, network products in support of synchronization typically conform to one or more IEEE 1588-2008 profiles. Each profile specifies how IEEE 1588-2008 is used in a given industry (e.g. telecom, automotive) and application. A profile can require features that are optional in IEEE 1588-2008, and it can specify new features that use IEEE 1588-2008 as a foundation.

It is expected that the IEEE 1588-2008 YANG module be used as follows:

o The IEEE 1588-2008 YANG module can be used as-is for products that conform to one of the default profiles specified in IEEE 1588-2008.

o When the IEEE 1588 standard is revised (e.g. the IEEE 1588 revision in progress scheduled to be published in 2017), it will add some new optional features to its data sets. The YANG module of this document can be revised and extended to add the new features (e.g. of IEEE 1588-2017). The YANG "revision" can be used to indicate changes to the YANG module.

o A profile standard based on IEEE 1588-2008 may create a dedicated YANG module for its profile. The profile's YANG module may use YANG "import" to import the IEEE 1588-2008 YANG module as its foundation. Then the profile's YANG module can use YANG "augment" to add any profile-specific enhancements.

o A product that conforms to a profile standard can also create its own YANG module. The product's YANG module can "import" the profile's module, and then use YANG "augment" to add any product-specific enhancements.

1.1. Conventions used in this document

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. Terminology

Most terminologies used in this document are extracted from [IEEE1588].

BC Boundary Clock

DS Data Set

E2E End-to-End

EUI Extended Unique Identifier.

GPS Global Positioning System

IANA Internet Assigned Numbers Authority

IP Internet Protocol

NIST National Institute of Standards and Technology

NTP Network Time Protocol

OC Ordinary Clock

P2P Peer-to-Peer

PTP Precision Time Protocol

TAI International Atomic Time

TC Transparent Clock

UTC Coordinated Universal Time

PTP dataset
Structured attributes of clocks (an OC, BC or TC) used for PTP protocol decisions and for providing values for PTP message fields, see Section 8 of [IEEE1588].

PTP instance
A PTP implementation in the device (i.e., an OC or BC) represented by a specific PTP dataset.

2. IEEE 1588-2008 YANG Model hierarchy

This section describes the hierarchy of an IEEE 1588-2008 YANG module. Query and configuration of device wide or port specific configuration information and clock data set are described for this version.

Query and configuration of clock information include:

(Note: The attribute names are consistent with IEEE 1588-2008, but changed to the YANG style, i.e., using all lower-case, with dashes between words.)

- Clock data set attributes in a clock node, including: current-ds, parent-ds, default-ds, time-properties-ds, and transparent-clock-default-ds.

- Port-specific data set attributes, including: port-ds and transparent-clock-port-ds.

The readers are assumed to be familiar with IEEE 1588-2008. As all PTP terminologies and PTP data set attributes are described in details in IEEE 1588-2008 [IEEE1588], this document only outlines each of them in the YANG module.

A simplified graphical representation of the data model is typically used by YANG modules as described in [RFC8040]. This document uses the same representation and 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 data (read-write) and "ro" state data (read-only). For IEEE 1588-2008, all data nodes are marked "rw" (see 2.2).

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.

o Arrow ("->") stands for a reference to a particular leaf instance in the tree.

```
module: ietf-ptp
  +--rw ptp
     +--rw instance-list* [instance-number]
     |  +--rw instance-number      uint32
     |  +--rw default-ds
     |  |  +--rw two-step-flag?    boolean
     |  |  +--rw clock-identity?   clock-identity-type
     |  |  +--rw number-ports?     uint16
     |  |  +--rw clock-quality
     |  |  |  +--rw clock-class?                 uint8
     |  |  |  +--rw clock-accuracy?              uint8
     |  |  |  +--rw offset-scaled-log-variance?   uint16
     |  |  +--rw priority1?        uint8
     |  |  +--rw priority2?        uint8
     |  |  +--rw domain-number?    uint8
     |  |  +--rw slave-only?       boolean
     |  +--rw current-ds
     |  |  +--rw steps-removed?       uint16
     |  |  +--rw offset-from-master?   time-interval-type
```

```
      |  |  +--rw mean-path-delay?      time-interval-type
      |  +--rw parent-ds
      |  |  +--rw parent-port-identity
      |  |  |  +--rw clock-identity?   clock-identity-type
      |  |  |  +--rw port-number?      uint16
      |  |  +--rw parent-stats?                     boolean
      |  |  +--rw observed-parent-offset-scaled-log-variance? uint16
      |  |  +--rw observed-parent-clock-phase-change-rate?   int32
      |  |  +--rw grandmaster-identity?            binary
      |  |  +--rw grandmaster-clock-quality
      |  |  |  +--rw clock-class?                  uint8
      |  |  |  +--rw clock-accuracy?               uint8
      |  |  |  +--rw offset-scaled-log-variance?   uint16
      |  |  +--rw grandmaster-priority1?           uint8
      |  |  +--rw grandmaster-priority2?           uint8
      |  +--rw time-properties-ds
      |  |  +--rw current-utc-offset-valid?   boolean
      |  |  +--rw current-utc-offset?         int16
      |  |  +--rw leap59?                     boolean
      |  |  +--rw leap61?                     boolean
      |  |  +--rw time-traceable?             boolean
      |  |  +--rw frequency-traceable?        boolean
      |  |  +--rw ptp-timescale?              boolean
      |  |  +--rw time-source?                uint8
      |  +--rw port-ds-list* [port-number]
      |     +--rw port-number               uint16
      |     +--rw port-state?               port-state-enumeration
      |     +--rw underlying-interface?         if:interface-ref
      |     +--rw log-min-delay-req-interval?   int8
      |     +--rw peer-mean-path-delay?         time-interval-type
      |     +--rw log-announce-interval?        int8
      |     +--rw announce-receipt-timeout?     uint8
      |     +--rw log-sync-interval?            int8
      |     +--rw delay-mechanism?       delay-mechanism-enumeration
      |     +--rw log-min-pdelay-req-interval?   int8
      |     +--rw version-number?                uint8
      +--rw transparent-clock-default-ds
      |  +--rw clock-identity?    clock-identity-type
      |  +--rw number-ports?      uint16
      |  +--rw delay-mechanism?   delay-mechanism-enumeration
      |  +--rw primary-domain?    uint8
      +--rw transparent-clock-port-ds-list* [port-number]
         +--rw port-number                     uint16
         +--rw clock-identity?                 clock-identity-type
         +--rw log-min-pdelay-req-interval?   int8
         +--rw faulty-flag?                    boolean
         +--rw peer-mean-path-delay?           time-interval-type
```

2.1. Interpretations from IEEE 1588 Working Group

The preceding model and the associated YANG module have some subtle differences from the data set specifications of IEEE Std 1588-2008. These differences are based on interpretation from the IEEE 1588 Working Group, and are intended to provide compatibility with future revisions of the IEEE 1588 standard.

In IEEE Std 1588-2008, a physical product can implement multiple PTP clocks (i.e. ordinary, boundary, or transparent clock). As specified in 1588-2008 subclause 7.1, each of the multiple clocks operates in an independent domain. However, the organization of multiple PTP domains was not clear in the data sets of IEEE Std 1588-2008. This document introduces the concept of PTP instance as described in the new revision of IEEE 1588. The instance concept is used exclusively to allow for optional support of multiple domains. The instance number has no usage within PTP messages.

Based on statements in IEEE 1588-2008 subclauses 8.3.1 and 10.1, most transparent clock products have interpreted the transparent clock data sets to reside as a singleton at the root level of the managed product, and this YANG model reflects that location.

2.2. Configuration and state

The information model of IEEE Std 1588-2008 classifies each member in PTP data sets as one of the following:

- Configurable: Writable by management.

- Dynamic: Read-only to management, and the value is changed by 1588 protocol operation.

- Static: Read-only to management, and the value typically does not change.

Under certain circumstances, the classification of an IEEE 1588 data set member can change. For details on the classification of each PTP data set member, refer to the IEEE Std 1588-2008 specification for that member.

3. IEEE 1588-2008 YANG Module

This module imports typedefs from [RFC7223]. Most attribute names are based on the information model defined in [IEEE1588], but adapted to the YANG style of naming.

```
<CODE BEGINS> file "ietf-ptp@2017-11-28.yang"

module ietf-ptp {
  namespace "urn:ietf:params:xml:ns:yang:ietf-ptp";
  prefix "ptp";

  import ietf-interfaces {
    prefix if;
  }

  organization "IETF TICTOC Working Group";
  contact
      "WG Web:   http://tools.ietf.org/wg/tictoc/
       WG List:  <mailto:tictoc@ietf.org>
       WG Chair: Karen O'Donoghue
                 <mailto:odonoghue@isoc.org>
       WG Chair: Yaakov Stein
                 <mailto: Yaakov_s@rad.com>
       Editor:   Yuanlong Jiang
                 <mailto:jiangyuanlong@huawei.com>
       Editor:   Rodney Cummings
                 <mailto:rodney.cummings@ni.com>";
  description
    "This YANG module defines a data model for the configuration
     of IEEE 1588-2008 clocks, and also for retrieval of the state
     data of IEEE 1588-2008 clocks.";

  revision "2017-11-28" {
    description "Version 7.0";
    reference "draft-ietf-tictoc-1588v2-yang";
  }

  typedef delay-mechanism-enumeration {
    type enumeration {
      enum e2e {
        value 1;
        description
          "The port uses the delay request-response mechanism.";
      }
      enum p2p {
        value 2;
```

```
          description
            "The port uses the peer delay mechanism.";
        }
        enum disabled {
          value 254;
          description
            "The port does not implement any delay mechanism.";
        }
      }
      description
        "The propagation delay measuring option used by the
         port. Values for this enumeration are specified
         by the IEEE 1588 standard exclusively.";
      reference
        "IEEE Std 1588-2008: 8.2.5.4.4";
    }

    typedef port-state-enumeration {
      type enumeration {
        enum initializing {
          value 1;
          description
            "The port is initializing its data sets, hardware, and
             communication facilities.";
        }
        enum faulty {
          value 2;
          description
            "The port is in the fault state.";
        }
        enum disabled {
          value 3;
          description
            "The port is disabled, and is not communicating PTP
             messages (other than possibly PTP management
             messages).";
        }
        enum listening {
          value 4;
          description
            "The port is listening for an Announce message.";
        }
        enum pre-master {
          value 5;
          description
            "The port is in the pre-master state.";
        }
```

```
        enum master {
          value 6;
          description
            "The port is behaving as a master port.";
        }
        enum passive {
          value 7;
          description
            "The port is in the passive state.";
        }
        enum uncalibrated {
          value 8;
          description
            "A master port has been selected, but the port is still
             in the uncalibrated state.";
        }
        enum slave {
          value 9;
          description
            "The port is synchronizing to the selected master port.";
        }
      }

      description
        "The current state of the protocol engine associated
         with the port.  Values for this enumeration are specified
         by the IEEE 1588 standard exclusively.";
      reference
        "IEEE Std 1588-2008: 8.2.5.3.1, 9.2.5";
    }

    typedef time-interval-type {
      type int64;
      description
        "Derived data type for time interval, represented in units of
          nanoseconds and multiplied by 2^16";
      reference
        "IEEE Std 1588-2008: 5.3.2";
    }

    typedef clock-identity-type {
      type binary {
        length "8";
      }
      description
        "Derived data type to identify a clock";
      reference
```

```
        "IEEE Std 1588-2008: 5.3.4";
    }

    grouping clock-quality-grouping {
      description
        "Derived data type for quality of a clock, which contains
         clockClass, clockAccuracy and offsetScaledLogVariance.";
      reference
        "IEEE Std 1588-2008: 5.3.7";

      leaf clock-class {
        type uint8;
        default 248;
        description
          "The clockClass denotes the traceability of the time
           or frequency distributed by the clock.";
      }

      leaf clock-accuracy {
        type uint8;
        description
          "The clockAccuracy indicates the expected accuracy
           of the clock.";
      }

      leaf offset-scaled-log-variance {
        type uint16;
        description
          "The offsetScaledLogVariance provides an estimate of
           the variations of the clock from a linear timescale
           when it is not synchronized to another clock
           using the protocol.";
      }
    }


    container ptp {
      description
        "The PTP struct containing all attributes of PTP Dataset,
          other optional PTP attributes can be augmented as well.";

      list instance-list {

        key "instance-number";

        description
```

```
          "List of one or more PTP datasets in the device (see IEEE
           Std 1588-2008 subclause 6.3).
           Each PTP dataset represents a distinct instance of
           PTP implementation in the device (i.e. distinct
           Ordinary Clock or Boundary Clock).";

        leaf instance-number {
          type uint32;
          description
             "The instance number of the current PTP instance.
              This instance number is used for management purposes
              only. This instance number does not represent the PTP
              domain number, and is not used in PTP messages.";
        }

        container default-ds {
          description
            "The default data set of the clock (see IEEE Std
             1588-2008 subclause 8.2.1).";

          leaf two-step-flag {
            type boolean;
            description
              "When set, the clock is a two-step clock; otherwise,
               the clock is a one-step clock.";
          }

          leaf clock-identity {
            type clock-identity-type;
            description
              "The clockIdentity of the local clock";
          }

          leaf number-ports {
            type uint16;
            description
              "The number of PTP ports on the instance.";
          }

          container clock-quality {
            description
              "The clockQuality of the local clock.";

            uses clock-quality-grouping;
          }

          leaf priority1 {
```

```
          type uint8;
          description
            "The priority1 attribute of the local clock.";
        }

        leaf priority2{
          type uint8;
          description
            "The priority2 attribute of the local clock. ";
        }

        leaf domain-number {
          type uint8;
          description
            "The domain number of the current syntonization
             domain.";
        }

        leaf slave-only {
          type boolean;
          description
            "When set, the clock is a slave-only clock.";
        }

      }

      container current-ds {
        description
          "The current data set of the clock (see IEEE Std
           1588-2008 subclause 8.2.2).";

        leaf steps-removed {
          type uint16;
          default 0;
          description
            "The number of communication paths traversed
             between the local clock and the grandmaster clock.";
        }

        leaf offset-from-master {
          type time-interval-type;
          description
            "The current value of the time difference between
             a master and a slave clock as computed by the slave.";
        }

        leaf mean-path-delay {
```

```
            type time-interval-type;
            description
              "The current value of the mean propagation time between
               a master and a slave clock as computed by the slave.";

          }

        }

        container parent-ds {
          description
            "The parent data set of the clock (see IEEE Std 1588-2008
             subclause 8.2.3).";

          container parent-port-identity {
            description
              "The portIdentity of the port on the master, it
               contains two members: clockIdentity and portNumber.";
            reference
               "IEEE Std 1588-2008: 5.3.5";

            leaf clock-identity {
              type clock-identity-type;
              description
                "Identity of the clock";
            }

            leaf port-number {
              type uint16;
              description
                "Port number";
            }
          }

          leaf parent-stats {
            type boolean;
            default false;
            description
              "When set, the values of
               observedParentOffsetScaledLogVariance and
               observedParentClockPhaseChangeRate of parentDS
               have been measured and are valid.";
          }

          leaf observed-parent-offset-scaled-log-variance {
            type uint16;
            default 65535;
```

```
          description
            "An estimate of the parent clock's PTP variance
             as observed by the slave clock.";
        }

        leaf observed-parent-clock-phase-change-rate {
          type int32;
          description
            "An estimate of the parent clock's phase change rate
             as observed by the slave clock.";
        }

        leaf grandmaster-identity {
          type binary {
            length "8";
          }
          description
            "The clockIdentity attribute of the grandmaster clock.";
        }

        container grandmaster-clock-quality {
          description
            "The clockQuality of the grandmaster clock.";
          uses clock-quality-grouping;
        }

        leaf grandmaster-priority1 {
          type uint8;
          description
            "The priority1 attribute of the grandmaster clock.";
        }

        leaf grandmaster-priority2 {
          type uint8;
          description
            "The priority2 attribute of the grandmaster clock.";
        }

      }

      container time-properties-ds {
        description
          "The timeProperties data set of the clock (see
           IEEE Std 1588-2008 subclause 8.2.4).";

        leaf current-utc-offset-valid {
          type boolean;
```

```
            description
              "When set, the current UTC offset is valid.";
          }

          leaf current-utc-offset {
            type int16;
            description
              "The offset between TAI and UTC when the epoch of the
               PTP system is the PTP epoch, i.e., when ptp-timescale
               is TRUE; otherwise, the value has no meaning.";
          }

          leaf leap59 {
            type boolean;
            description
              "When set, the last minute of the current UTC day
               contains 59 seconds.";
          }

          leaf leap61 {
            type boolean;
            description
              "When set, the last minute of the current UTC day
               contains 61 seconds.";
          }

          leaf time-traceable {
            type boolean;
            description
              "When set, the timescale and the currentUtcOffset are
               traceable to a primary reference.";
          }

          leaf frequency-traceable {
            type boolean;
            description
              "When set, the frequency determining the timescale
               is traceable to a primary reference.";
          }

          leaf ptp-timescale {
            type boolean;
            description
              "When set, the clock timescale of the grandmaster
               clock is PTP; otherwise, the timescale is ARB
               (arbitrary).";
          }
```

```
        leaf time-source {
          type uint8;
          description
            "The source of time used by the grandmaster clock.";
        }
      }

      list port-ds-list {
        key "port-number";
        description
          "List of port data sets of the clock (see IEEE Std
           1588-2008 subclause 8.2.5).";

        leaf port-number{
          type uint16;
          description
            "Port number.
             The data sets (i.e. information model) of IEEE Std
             1588-2008 specify a member portDS.portIdentity, which
             uses a typed struct with members clockIdentity and
             portNumber.

             In this YANG data model, portIdentity is not modeled
             in the port-ds-list, however, its members are provided
             as follows:
             portIdentity.portNumber is provided as this port-
             number leaf in port-ds-list; and
             portIdentity.clockIdentity is provided as the clock-
             identity leaf in default-ds of the instance
             (i.e. ../../default-ds /clock-identity).";
        }

        leaf port-state {
          type port-state-enumeration;
          default "initializing";
          description
            "Current state associated with the port.";
        }

        leaf underlying-interface {
         type if:interface-ref;
         description
            "Reference to the configured underlying interface that
             is used by this PTP Port (see RFC 7223).";
        }
```

```
          leaf log-min-delay-req-interval {
            type int8;
            description
              "The base-two logarithm of the minDelayReqInterval
               (the minimum permitted mean time interval between
               successive Delay_Req messages).";
          }

          leaf peer-mean-path-delay {
            type time-interval-type;
            default 0;
            description
              "An estimate of the current one-way propagation delay
               on the link when the delayMechanism is P2P; otherwise,
               it is zero.";
          }

          leaf log-announce-interval {
            type int8;
            description
              "The base-two logarithm of the mean
               announceInterval (mean time interval between
               successive Announce messages).";
          }

          leaf announce-receipt-timeout {
            type uint8;
            description
              "The number of announceInterval that have to pass
               without receipt of an Announce message before the
               occurrence of the event ANNOUNCE_RECEIPT_TIMEOUT_
               EXPIRES.";
          }

          leaf log-sync-interval {
            type int8;
            description
              "The base-two logarithm of the mean SyncInterval
               for multicast messages.  The rates for unicast
               transmissions are negotiated separately on a per port
               basis and are not constrained by this attribute.";
          }

          leaf delay-mechanism {
            type delay-mechanism-enumeration;
            description
              "The propagation delay measuring option used by the
```

```
                port in computing meanPathDelay.";
           }

           leaf log-min-pdelay-req-interval {
             type int8;
             description
               "The base-two logarithm of the
                minPdelayReqInterval (minimum permitted mean time
                interval between successive Pdelay_Req messages).";

           }

           leaf version-number {
             type uint8;
             description
               "The PTP version in use on the port.";
           }

         }
      }

      container transparent-clock-default-ds {
        description
          "The members of the transparentClockDefault Data Set (see
           IEEE Std 1588-2008 subclause 8.3.2).";

        leaf clock-identity {
          type clock-identity-type;
          description
            "The clockIdentity of the transparent clock.";
        }

        leaf number-ports {
          type uint16;
          description
            "The number of PTP ports on the Transparent Clock.";
        }

        leaf delay-mechanism {
          type delay-mechanism-enumeration;
          description
            "The propagation delay measuring option
             used by the transparent clock.";
        }

        leaf primary-domain {
          type uint8;
```

```
          default 0;
          description
            "The domainNumber of the primary syntonization domain.";
        }
      }
      list transparent-clock-port-ds-list {
        key "port-number";
        description
          "List of transparentClockPort data sets of the transparent
           clock (see IEEE Std 1588-2008 subclause 8.3.3).";

        leaf port-number {
            type uint16;
            description
              "Port number.
               The data sets (i.e. information model) of IEEE Std
               1588-2008 specify a member
               transparentClockPortDS.portIdentity, which uses a typed
               struct with members clockIdentity and portNumber.

               In this YANG data model, portIdentity is not modeled in
               the transparent-clock-port-ds-list, however,
               portIdentity.portNumber is provided as this leaf member
               in transparent-clock-port-ds-list.";

        }

        leaf clock-identity {
            type clock-identity-type;
            description
              "clock-identity.
               The data sets (i.e. information model) of IEEE Std
               1588-2008 specify a member
               transparentClockPortDS.portIdentity, which uses a typed
               struct with members clockIdentity and portNumber.

               In this YANG data model, portIdentity is not modeled in
               the transparent-clock-port-ds-list, however,
               portIdentity.clockIdentity is provided as this leaf
               member in transparent-clock-port-ds-list.";
        }

        leaf log-min-pdelay-req-interval {
          type int8;
          description
            "The logarithm to the base 2 of the
             minPdelayReqInterval (minimum permitted mean time
```

```
             interval between successive Pdelay_Req messages).";
        }

        leaf faulty-flag {
          type boolean;
          default false;
          description
            "When set, the port is faulty.";
        }

        leaf peer-mean-path-delay {
          type time-interval-type;
          default 0;
          description
            "An estimate of the current one-way propagation delay
             on the link when the delayMechanism is P2P; otherwise,
             it is zero.";
        }

       }
     }
   }

   <CODE ENDS>
```

4. Security Considerations

YANG modules are designed to be accessed via the NETCONF protocol [RFC6241], thus security considerations in [RFC6241] apply here. Security measures such as using the NETCONF over SSH [RFC6242] and restricting its use with access control [RFC6536] can further improve its security, avoid injection attacks and misuse of the protocol. Furthermore, general security considerations of time protocols are discussed in [RFC7384].

Some data nodes defined in this YANG module are writable, and an inappropriate use of them may adversely impact a synchronization network. For example, loss of synchronization on a clock, accuracy degradation on a set of clocks, or even break down of a whole synchronization network.

5. IANA Considerations

This document registers a URI in the IETF XML registry, and the following registration is requested to be made:
URI: urn:ietf:params:xml:ns:yang:ietf-ptp

This document registers a YANG module in the YANG Module Names:
name: ietf-ptp namespace: urn:ietf:params:xml:ns:yang:ietf-ptp

6. References

6.1. Normative References

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

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

[RFC6991] Schoenwaelder, J., "Common YANG Data Types", RFC 6991, July 2013

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, May 2014

[IEEE1588] IEEE, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Std 1588-2008, July 2008

6.2. Informative References

[IEEE8021AS] IEEE, "Timing and Synchronizations for Time-Sensitive Applications in Bridged Local Area Networks", IEEE 802.1AS-2001, 2011

[RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, January 2003

[RFC4663] Harrington, D., "Transferring MIB Work from IETF Bridge MIB WG to IEEE 802.1 WG", RFC 4663, September 2006

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

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

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, March 2012

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, October 2014

[RFC8040] Bierman, A., Bjorklund, M., and Watsen, K., "RESTCONF protocol", RFC 8040, January 2017

[RFC8173] Shankarkumar, V., Montini, L., Frost, T., and Dowd, G., "Precision Time Protocol Version 2 (PTPv2) Management Information Base", RFC 8173, June 2017

7. Acknowledgments

The authors would like to thank Joe Gwinn, Mahesh Jethanandani, Tal Mizrahi, Opher Ronen, Liang Geng, Alex Campbell, Tom Petch and John Fletcher for their valuable reviews and suggestions, and thank Benoit Claise and Radek Krejci for their validation of the YANG module.

Appendix A Transferring YANG Work to IEEE 1588 WG (Informational)

This appendix describes a future plan to transition responsibility for IEEE 1588 YANG modules from the IETF TICTOC Working Group (WG) to the IEEE 1588 WG, which develops the time synchronization technology that the YANG modules are designed to manage.

This appendix is forward-looking with regard to future standardization roadmaps in IETF and IEEE. Since those roadmaps cannot be predicted with significant accuracy, this appendix is informational, and it does not specify imperatives or normative specifications of any kind.

The IEEE 1588-2008 YANG module of this standard represents a cooperation between IETF (for YANG) and IEEE (for 1588). For the initial standardization of IEEE-1588 YANG modules, the information model is relatively clear (i.e. IEEE 1588 data sets), but expertise in YANG is required, making IETF an appropriate location for the standards. The TICTOC WG has expertise with IEEE 1588, making it the appropriate location within IETF.

The IEEE 1588 WG anticipates future changes to its standard on an ongoing basis. As IEEE 1588 WG members gain practical expertise with YANG, the IEEE 1588 WG will become more appropriate for standardization of its YANG modules. As the IEEE 1588 standard is revised and/or amended, IEEE 1588 members can more effectively synchronize the revision of this YANG module with future versions of the IEEE 1588 standard.

This appendix is meant to establish some clear expectations between IETF and IEEE about the future transfer of IEEE 1588 YANG modules to the IEEE 1588 WG. The goal is to assist in making the future transfer as smooth as possible. As the transfer takes place, some case-by-case situations are likely to arise, which can be handled by discussion on the IETF TICTOC WG mailing lists and/or appropriate liaisons.

This appendix obtained insight from [RFC4663], an informational memo that described a similar transfer of MIB work from the IETF Bridge MIB WG to the IEEE 802.1 WG.

A.1. Assumptions for the Transfer

For the purposes of discussion in this appendix, assume that the IETF TICTOC WG has approved a standard YANG module for a published IEEE 1588 standard. As of this writing, this is IEEE Std 1588-2008,

but it is possible that YANG for subsequent 1588 revisions could be published from the IETF TICTOC WG. For discussion in this appendix, we use the phrase "last IETF 1588 YANG" to refer to most recently published 1588 YANG from the IETF TICTOC WG.

The IEEE-SA Standards Board New Standards Committee (NesCom) handles new Project Authorization Requests (PARs) (see http://standards.ieee.org/board/nes/). PARs are roughly the equivalent of IETF Working Group Charters and include information concerning the scope, purpose, and justification for standardization projects.

Assume that IEEE 1588 has an approved PAR that explicitly specifies development of a YANG module. The transfer of YANG work will occur in the context of this IEEE 1588 PAR. For discussion in this appendix, we use the phrase "first IEEE 1588 YANG" to refer to the first IEEE 1588 standard for YANG.

Assume that as part of the transfer of YANG work, the IETF TICTOC WG agrees to cease all work on standard YANG modules for IEEE 1588.

Assume that the IEEE 1588 WG has participated in the development of the last IETF 1588 YANG module, such that the first IEEE 1588 YANG module will effectively be a revision of it. In other words, the transfer of YANG work will be relatively clean.

The actual conditions for the future transfer can be such that the preceding assumptions do not hold. Exceptions to the assumptions will need to be addressed on a case-by-case basis at the time of the transfer. This appendix describes topics that can be addressed based on the preceding assumptions.

A.2. Intellectual Property Considerations

During review of the legal issues associated with transferring Bridge MIB WG documents to the IEEE 802.1 WG (Section 3.1 and Section 9 of [RFC4663]), it was concluded that the IETF does not have sufficient legal authority to make the transfer to IEEE without the consent of the document authors.

If the last IETF 1588 YANG is published as a RFC, the work is required to be transferred from the IETF to the IEEE, so that IEEE 1588 WG can begin working on the first IEEE 1588 YANG.

When work on the first IEEE YANG module begins in the IEEE 1588 WG, that work derives from the last IETF YANG module of this RFC, requiring a transfer of that work from the IETF to the IEEE. In

order to avoid having the transfer of that work be dependent on the availability of this RFC's authors at the time of its publication, the IEEE Standards Association department of Risk Management and Licensing provided the appropriate forms and mechanisms for this document's authors to assign a non-exclusive license for IEEE to create derivative works from this document. Those IEEE forms and mechanisms will be updated as needed during the development of this document and any future IETF YANG modules for IEEE 1588. This will help to make the future transfer of work from IETF to IEEE occur as smoothly as possible.

As stated in the initial "Status of this Memo", the YANG module in this document conforms to the provisions of BCP 78. The IETF will retain all the rights granted at the time of publication in the published RFCs.

A.3. Namespace and Module Name

As specified in the "IANA Considerations" section, the YANG module in this document uses IETF as the root of its URN namespace and YANG module name.

Use of IETF as the root of these names implies that the YANG module is standardized in a Working Group of IETF, using the IETF processes. If the IEEE 1588 Working Group were to continue using these names rooted in IETF, the IEEE 1588 YANG standardization would need to continue in the IETF. The goal of transferring the YANG work is to avoid this sort of dependency between standards organizations.

IEEE 802 has an active PAR (IEEE P802d) for creating a URN namespace for IEEE use (see http://standards.ieee.org/develop/project/802d.html). It is likely that this IEEE 802 PAR will be approved and published prior to the transfer of YANG work to the IEEE 1588 WG. If so, the IEEE 1588 WG can use the IEEE URN namespace for the first IEEE 1588 YANG module, such as:

    urn:ieee:Std:1588:yang:ieee1588-ptp

where "ieee1588-ptp" is the registered YANG module name in the IEEE.

Under the assumptions of section A.1, the first IEEE 1588 YANG module prefix can be the same as the last IETF 1588 YANG module prefix (i.e. "ptp"), since the nodes within both YANG modules are compatible.

The result of these name changes are that for complete compatibility, a server (i.e. IEEE 1588 node) can choose to implement a YANG module for the last IETF 1588 YANG module (with IETF root) as well as the first IEEE 1588 YANG module (with IEEE root). Since the content of the YANG module transferred are the same, the server implementation is effectively common for both.

From a client's perspective, a client of the last IETF 1588 YANG module (or earlier) looks for the IETF-rooted module name; and a client of the first IEEE 1588 YANG module (or later) looks for the IEEE-rooted module name.

A.4. IEEE 1588 YANG Modules in ASCII Format

Although IEEE 1588 can certainly decide to publish YANG modules only in the PDF format that they use for their standard documents, without publishing an ASCII version, most network management systems cannot import the YANG module directly from the PDF. Thus, not publishing an ASCII version of the YANG module would negatively impact implementers and deployers of YANG modules and would make potential IETF reviews of YANG modules more difficult.

This appendix recommends that the IEEE 1588 WG consider future plans for:

- o Public availability of the ASCII YANG modules during project development. These ASCII files allow IETF participants to access these documents for pre-standard review purposes.

- o Public availability of the YANG portion of published IEEE 1588 standards, provided as an ASCII file for each YANG module. These ASCII files are intended for use of the published IEEE 1588 standard.

As an example of public availability during project development, IEEE 802 uses the same repository that IETF uses for YANG module development (see https://github.com/YangModels/yang). IEEE branches are provided for experimental work (i.e. pre-PAR) as well as standard work (post-PAR drafts). IEEE-SA has approved use of this repository for project development, but not for published standards.

As an example of public availability of YANG modules for published standards, IEEE 802.1 provides a public list of ASCII files for MIB (see http://www.ieee802.org/1/files/public/MIBs/ and http://www.ieee802.org/1/pages/MIBS.html), and analogous lists are planned for IEEE 802.1 YANG files.

Authors' Addresses

Yuanlong Jiang (Editor)
Huawei Technologies Co., Ltd.
Bantian, Longgang district
Shenzhen 518129, China
Email: jiangyuanlong@huawei.com

Xian Liu
Independent
Shenzhen 518129, China
lene.liuxian@foxmail.com

Jinchun Xu
Huawei Technologies Co., Ltd.
Bantian, Longgang district
Shenzhen 518129, China
xujinchun@huawei.com

Rodney Cummings (Editor)
National Instruments
11500 N. Mopac Expwy
Bldg. C
Austin, TX 78759-3504
Email: Rodney.Cummings@ni.com