Internet Engineering Task Force (IETF)
Request for Comments: 8575
Category: Standards Track
ISSN: 2070-1721
May 2019

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

YANG Data Model for the Precision Time Protocol (PTP)

Abstract

This document defines a YANG data model for the configuration of devices and clocks using the Precision Time Protocol (PTP) as specified in IEEE Std 1588-2008. It also defines the retrieval of the configuration information, the data sets and the running states of PTP clocks. The YANG module in this document conforms to the Network Management Datastore Architecture (NMDA).

Status of This Memo

Table of Contents

1. Introduction

As a synchronization protocol, IEEE Std 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 nanoseconds. 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 Std 1588-2008 are numerous.

According to the concepts described in [RFC3444], IEEE Std 1588-2008 itself provides an information model in its normative specifications for the data sets (in IEEE Std 1588-2008 clause 8). Some standardization organizations, including the IETF, have specified data models in MIBs (Management Information Bases) for IEEE Std 1588-2008 data sets (e.g., [RFC8173] and [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 Std 1588-2008 synchronization network be flexible and more Internet based (typically overlaid on their transport networks). Software-Defined Networking (SDN) is another driving factor, which demands an improved configuration capability of synchronization networks.

YANG [RFC7950] 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 is defined in [RFC7950]; a collection of common data types is also defined in [RFC6991]. Advantages of YANG include Internet-based configuration capabilities, validation, rollback, and so on. All of these characteristics make it attractive to become another candidate modeling language for IEEE Std 1588-2008.

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

The YANG module in this document conforms to the Network Management Datastore Architecture (NMDA) [RFC8342].

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

The readers are assumed to be familiar with IEEE Std 1588-2008. It is expected that the IEEE Std 1588-2008 YANG module will be used as follows:

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

- When the IEEE Std 1588 standard is revised (e.g., the IEEE Std 1588 revision in progress at the time of writing this document), it will add some new optional features to its data sets. The YANG module of this document can be revised and extended to support these new features. Moreover, the YANG "revision" MUST be used to indicate changes to the YANG module under such a circumstance.

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

- A product that conforms to a profile standard may also create its own YANG module. The product's YANG module SHOULD "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", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Terminology

Most terminology used in this document is extracted from [IEEE1588].

BC Boundary Clock, see Section 3.1.3 of [IEEE1588]

DS Data Set, see Section 8.1.1 of [IEEE1588]

E2E End-to-End, see Section 3.2 of [IEEE1588]

IANA Internet Assigned Numbers Authority

OC Ordinary Clock, see Section 3.1.22 of [IEEE1588]

P2P Peer-to-Peer, see Section 3.2 of [IEEE1588]

PTP Precision Time Protocol, see Section 3.1.28 of [IEEE1588]

TAI International Atomic Time, see Section 3.2 of [IEEE1588]

TC Transparent Clock, see Section 3.1.46 of [IEEE1588]

UTC Coordinated Universal Time, see Section 3.2 of [IEEE1588]

PTP data set
Structured attributes of clocks (an OC, BC, or TC) used for PTP 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 data set.

2. IEEE Std 1588-2008 YANG Data Model Hierarchy

This section describes the hierarchy of a YANG module for IEEE Std 1588-2008; specifically, query and configuration of device-wide or port-specific configuration information and clock data sets are described.

Query and configuration of clock information include:

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

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

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

As all PTP terminology and PTP data set attributes are described in detail in IEEE Std 1588-2008, this document only outlines each of them in the YANG module.

A simplified YANG tree diagram [RFC8340] representing the data model is typically used by YANG modules. This document uses the same tree diagram syntax as described in [RFC8340].

```
module: ietf-ptp
  +--rw ptp
     +--rw instance-list* [instance-number]
     |  +--rw instance-number      uint32
     |  +--rw default-ds
     |  |  +--rw two-step-flag?    boolean
     |  |  +--ro 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?          clock-identity-type
     |  |  +--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
           |  +--ro 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 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 they are intended to provide compatibility with future revisions of the IEEE Std 1588 standard.

In IEEE Std 1588-2008, a physical product can implement multiple PTP clocks (i.e., an ordinary, boundary, or transparent clock). As specified in IEEE Std 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 a PTP instance, which is a PTP implementation in a device (i.e., an OC or BC) represented by a specific PTP data set. Each instance operates in exactly one domain. 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 Std 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 data 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 PTP protocol operation.

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

For details on the classification of each PTP data set member, refer to the specification of that member in IEEE Std 1588-2008.

Under certain circumstances, the classification of an IEEE Std 1588 data set member may change for a YANG implementation, for example, a configurable member needs to be changed to read-only. In such a case, an implementation SHOULD choose to return a warning upon writing to a read-only member or use the deviation mechanism to develop a new deviation model as described in Section 7.20.3 of [RFC7950].

3. IEEE Std 1588-2008 YANG Module

This module imports typedef "interface-ref" from [RFC8343]. Most attributes are based on the information model defined in [IEEE1588], but their names are adapted to the YANG style of naming.

```
<CODE BEGINS> file "ietf-ptp@2019-05-07.yang"
module ietf-ptp {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ptp";
  prefix ptp;

  import ietf-interfaces {
    prefix if;
    reference
      "RFC 8343: A YANG Data Model for Interface Management";
  }

  organization
    "IETF TICTOC Working Group";
  contact
    "WG Web:   https://datatracker.ietf.org/wg/tictoc/
     WG List:  <mailto:tictoc@ietf.org>
     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 Std 1588-2008 clocks, and also for retrieval of the state
     data of IEEE Std 1588-2008 clocks.";

  revision 2019-05-07 {
    description
      "Initial version";
    reference
      "RFC 8575: YANG Data Model for the Precision Time Protocol";
  }

  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 Std 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 Std 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 data set,
          other optional PTP attributes can be augmented as well.";
      list instance-list {
        key "instance-number";
        description
          "List of one or more PTP data sets in the device (see IEEE
           Std 1588-2008 subclause 6.3).
           Each PTP data set 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).  This data set represents
             the configuration/state required for operation
             of Precision Time Protocol (PTP) state machines.";
          reference
            "IEEE Std 1588-2008: 8.2.1";
          leaf two-step-flag {
            type boolean;
            description
              "When set to true, the clock is a two-step clock;
               otherwise,the clock is a one-step clock.";
          }
          leaf clock-identity {
            type clock-identity-type;
            config false;
            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 to true, 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).  This data set represents
           local states learned from the exchange of
           Precision Time Protocol (PTP) messages.";
        reference
          "IEEE Std 1588-2008: 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).";
        reference
          "IEEE Std 1588-2008: 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 to true, 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 clock-identity-type;
        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).";
        reference
          "IEEE Std 1588-2008: 8.2.4";
        leaf current-utc-offset-valid {
          type boolean;
          description
            "When set to true, the current UTC offset is valid.";
        }
        leaf current-utc-offset {
          when "../current-utc-offset-valid='true'";
          type int16;
          description
            "The offset between TAI and UTC when the epoch of the
             PTP system is the PTP epoch in units of seconds, i.e.,
             when ptp-timescale is TRUE; otherwise, the value has
             no meaning.";
        }
        leaf leap59 {
          type boolean;
          description
            "When set to true, the last minute of the current UTC
             day contains 59 seconds.";
        }
        leaf leap61 {
          type boolean;
          description
            "When set to true, the last minute of the current UTC
             day contains 61 seconds.";
        }
        leaf time-traceable {
          type boolean;
          description
            "When set to true, the timescale and the
             currentUtcOffset are traceable to a primary
             reference.";
        }
        leaf frequency-traceable {
          type boolean;
          description
            "When set to true, the frequency determining the
             timescale is traceable to a primary reference.";
```

```
          }
          leaf ptp-timescale {
            type boolean;
            description
              "When set to true, 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).";
          reference
            "IEEE Std 1588-2008: 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 8343).";
          reference
            "RFC 8343: A YANG Data Model for Interface Management";
        }
        leaf log-min-delay-req-interval {
          type int8;
          description
            "The base-2 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-2 logarithm of the mean
             announceInterval (mean time interval between
             successive Announce messages).";
        }
        leaf announce-receipt-timeout {
          type uint8;
          description
            "The number of announceIntervals 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-2 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-2 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).";
      reference
        "IEEE Std 1588-2008: 8.3.2";
      leaf clock-identity {
        type clock-identity-type;
        config false;
        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 (see
           IEEE Std 1588-2008 subclause 10.1).";
```

```
          reference
            "IEEE Std 1588-2008: 10.1";
        }
      }
      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).";
        reference
          "IEEE Std 1588-2008: 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, its
             members are provided as follows:
             portIdentity.portNumber is provided as this leaf member
             in transparent-clock-port-ds-list and
             portIdentity.clockIdentity is provided as the
             clock-identity leaf in transparent-clock-default-ds
             (i.e., ../../transparent-clock-default-ds/clock-
             identity).";
        }
        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 to true, 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

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446]. Furthermore, general security considerations of time protocols are discussed in [RFC7384].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable, and the involved subtrees that are sensitive include:

/ptp/instance-list specifies an instance (i.e., PTP data sets) for an OC or BC.

/ptp/transparent-clock-default-ds specifies a default data set for a TC.

/ptp/transparent-clock-port-ds-list specifies a list of port data sets for a TC.

Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. Specifically, an inappropriate configuration of them may adversely impact a PTP 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 the following URI in the "IETF XML Registry" [RFC3688]:

URI: urn:ietf:params:xml:ns:yang:ietf-ptp
Registrant Contact: The IESG
XML: N/A; the requested URI is an XML namespace

This document registers the following YANG module in the "YANG Module Names" registry [RFC6020]:

Name: ietf-ptp
Namespace: urn:ietf:params:xml:ns:yang:ietf-ptp
Prefix: ptp
Reference: RFC 8575

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[IEEE1588] IEEE, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Std 1588-2008, DOI 10.1109/IEEESTD.2008.4579760, July 2008.

6.2. Informative References

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

[RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, DOI 10.17487/RFC3444, January 2003, <https://www.rfc-editor.org/info/rfc3444>.

[RFC4663] Harrington, D., "Transferring MIB Work from IETF Bridge MIB WG to IEEE 802.1 WG", RFC 4663, DOI 10.17487/RFC4663, September 2006, <https://www.rfc-editor.org/info/rfc4663>.

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384, October 2014, <https://www.rfc-editor.org/info/rfc7384>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8173] Shankarkumar, V., Montini, L., Frost, T., and G. Dowd, "Precision Time Protocol Version 2 (PTPv2) Management Information Base", RFC 8173, DOI 10.17487/RFC8173, June 2017, <https://www.rfc-editor.org/info/rfc8173>.

Appendix A. Transferring YANG Work to the IEEE 1588 WG

This Appendix is informational.

This appendix describes a future plan to transition responsibility for IEEE Std 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 the 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 Std 1588-2008 YANG module of this standard represents a cooperation between the 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 Std 1588 data sets), but expertise in YANG is required, making IETF an appropriate location for the standards. The TICTOC WG has expertise with IEEE Std 1588, making it the appropriate location within the 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 IESG has approved the publication of an RFC containing a YANG module for a published IEEE 1588 standard. As of this writing, this is IEEE Std 1588-2008, but it is possible that YANG modules 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 the most recently published 1588 YANG module 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 (Sections 3.1 and 9 of [RFC4663]), it was concluded that the IETF does not have sufficient legal authority to make the transfer to the IEEE without the consent of the document authors.

If the last IETF 1588 YANG is published as an 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 for any future IETF YANG modules for IEEE 1588 (the signed forms are held by the IEEE Standards Association department of Risk Management and Licensing.). This will help to make the future transfer of work from the IETF to the 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 Section 5 "IANA Considerations", 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 Appendix A.1, the first IEEE 1588 YANG module's prefix will be the same as the last IETF 1588 YANG module's prefix (i.e., "ptp"). Consequently, other YANG modules can preserve the same import prefix "ptp" to access PTP nodes during the migration from the last IETF 1588 YANG module to the first IEEE 1588 YANG module.

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:

- 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.

- 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.

Acknowledgments

The authors would like to thank Tom Petch, Radek Krejci, Mahesh Jethanandani, Tal Mizrahi, Opher Ronen, Liang Geng, Alex Campbell, Joe Gwinn, John Fletcher, William Zhao, and Dave Thaler for their valuable reviews and suggestions. They would like to thank Benoit Claise and Radek Krejci for their validation of the YANG module, and thank Jingfei Lv and Zitao Wang for their discussions on IEEE 1588 and YANG, respectively.

Authors' Addresses

Yuanlong Jiang (editor)
Huawei
Bantian, Longgang district
Shenzhen 518129
China

Email: jiangyuanlong@huawei.com

Xian Liu
Independent
Shenzhen 518129
China

Email: lene.liuxian@foxmail.com

Jinchun Xu
Huawei
Bantian, Longgang district
Shenzhen 518129
China

Email: xujinchun@huawei.com

Rodney Cummings (editor)
National Instruments
11500 N. Mopac Expwy Bldg. C
Austin, TX 78759-3504
United States of America

Email: Rodney.Cummings@ni.com