Deterministic Networking
Internet-Draft
Intended status: Standards Track
Expires: September 30, 2018

S. Ben Hadj Said
M. Boc
CEA, LIST

March 29, 2018

YANG Model of IEEE 802.1AS
draft-benhadjsaid-detnet-gptp-yang-00

Abstract

This document describes a YANG data model for the management of IEEE 802.1AS module in network devices. This data model includes configuration data and state data (status information and counters for the collection of statistics).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 30, 2018.

Table of Contents

1. Introduction

Multiple use cases described in [I-D.ietf-detnet-use-cases] have highlighted the need for Deterministic Networking (DetNet) service where an accurate time synchronization, low packet loss rates, and low and stable end-to-end latencies are ensured. According to [I-D.ietf-detnet-flow-information-model], one way to provide the DetNet service for Layer 2 flows is to use the Time-Sensitive Networking (TSN) mechanisms that were specified by IEEE 802.1 TSN working group [IEEE8021TSN] for a bridged or virtually bridged Local Area Networks. The documents [I-D.ietf-detnet-architecture] (see Section 4) and [I-D.ietf-detnet-flow-information-model] target to use the Centralized Network / Distributed user model out of the three models specified by [IEEE8021Qcc]. In this model, the network configuration is done via a central entity via a network management protocol (e.g. SNMP, NETCONF [RFC6241], etc.).

Time synchronization is a key feature for DetNet service. To ensure an accurate synchronization, the IEEE 802.1 TSN group is relying on the standard IEEE 802.1AS (generalized Precision Time Protocol gPTP)[IEEE8021AS]. This standard allows to synchronize clocks of the different Ethernet devices across bridged or virtually bridged Local Area Network. The principle of IEEE 802.1AS consists in electing, dynamically, one of the devices to be the grandmaster to whom the rest of devices synchronize their clocks thanks to the Best Master Clock Algorithm (BMCA). IEEE 802.1AS requires that several parameters to be configured with the same value in each device participating in the time synchronization domain. For example, the frequency with which the Sync messages are sent via each link should be the same in the whole gPTP domain; otherwise it may impact the time synchronization accuracy. In order to avoid such cases, the management/control plane entity shall be able to configure the IEEE

802.1AS module in each network node in data plane in an adequate manner.

The IEEE 802.1AS has specified its Management Information Base (IEEE8021-AS-MIB)[IEEE8021AS]. This kind of data model limit the management and control plane entity to use only SNMP protocol for IEEE 802.1AS configuration. Providing the IEEE 802.1AS YANG data model is crucial as it promotes interoperability between networking devices, centralized management entity (e.g. SDN controller, etc.) and applications from different manufacturers. For instance, several protocols such as NETCONF [RFC6241] and RESTCONF [I-D.draft-ietf-netconf-restconf] can use this YANG data model to get state information related to IEEE 802.1AS module or to configure some of its parameters.

This document describes the YANG data model [RFC6020] [RFC7950] of the IEEE 802.1AS module [IEEE8021AS].

The definition of ietf-gptp module is based the standard IEEE 802.1AS as well as its IEEE8021-AS-MIB module. The model has been validated with pyang tool and the on-line tool yangdump-pro[http://www.netconfcentral.org/run_yangdump].

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

MIB: Management Information Base

gPTP: generalized Precision Time Protocol, i.e. IEEE 802.1AS

YANG: Yet Another Next Generation

3. Tree Diagrams

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

- Brackets "[" and "]" enclose list keys.
- Abbreviations before data node names: "rw" means configuration(read-write) and "ro" state data (read-only).
- Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.

- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").

- Ellipsis ("...") stands for contents of subtrees that are not shown.

4. IEEE 802.1AS: Overview

The tree diagram of IEEE 802.1AS data model is depicted in Figure 1.

```
module: ietf-gptp
  +--rw defaultDataSet
  |  +--ro clockIdentity?           clock_identity_type
  |  +--ro numberPorts?             uint32
  |  +--ro clockQuality
  |  |  +--ro clockClass?                uint8
  |  |  +--ro clockAccuracy?             uint8
  |  |  +--ro offsetScaledLogVariance?   uint16
  |  +--rw priority1?               uint32
  |  +--rw priority2?               uint32
  |  +--ro gmCapable?               boolean
  |  +--rw currentUtcOffset?        uint32
  |  +--rw currentUtcOffsetValid?   boolean
  |  +--rw leap59?                  boolean
  |  +--rw leap61?                  boolean
  |  +--ro timeTraceable?           boolean
  |  +--ro frequencyTraceable?      boolean
  |  +--ro timeSource?              time-source-enumeration
  +--ro currentDataSet
  |  +--ro stepsRemoved?                   int32
  |  +--ro offsetFromMaster
  |  |  +--ro offsetFromMasterHs?   int32
  |  |  +--ro offsetFromMasterMs?   int32
  |  |  +--ro offsetFromMasterLs?   int32
  |  +--ro lastGmPhaseChange
  |  |  +--ro lastGmPhaseChangeHs?   uint32
  |  |  +--ro lastGmPhaseChangeMs?   uint32
  |  |  +--ro lastGmPhaseChangeLs?   uint32
  |  +--ro lastGmFreqChange
  |  |  +--ro lastGmFreqChangeMs?   int32
  |  |  +--ro lastGmFreqChangeLs?   int32
  |  +--ro gmTimebaseIndicator?            uint32
  |  +--ro gmChangeCount?                  yang:counter32
  |  +--ro timeOfLastGmChangeEvent?        yang:timestamp
  |  +--ro timeOfLastGmFreqChangeEvent?    yang:timestamp
  |  +--ro timeOfLastGmPhaseChangeEvent?   yang:timestamp
  +--ro parentDataSet
  |  +--ro parentPortIdentity
```

```
  |  |  +--ro clockIdentity?   clock_identity_type
  |  |  +--ro portNumber?      uint32
  |  +--ro cumulativeRateRatio?       int32
  |  +--ro grandmasterIdentity?       clock_identity_type
  |  +--ro grandmasterClockQuality
  |  |  +--ro clockClass?                uint8
  |  |  +--ro clockAccuracy?             uint8
  |  |  +--ro offsetScaledLogVariance?   uint16
  |  +--ro grandmasterPriority1?      uint32
  |  +--ro grandmasterPriority2?      uint32
  +--ro timePropertiesDataSet
  |  +--ro currentUTCOffset?        int32
  |  +--ro currentUTCOffsetValid?   boolean
  |  +--ro leap59?                  boolean
  |  +--ro leap61?                  boolean
  |  +--ro timeTraceable?           boolean
  |  +--ro frequencyTraceable?      boolean
  |  +--ro timeSource?              time-source-enumeration
  +--rw portDataSet* [portNumber]
  |  +--ro bridgeBasePort                    bridge-port-number
  |  +--ro asIfIndex                         if-index
  |  +--rw portNumber                        uint32
  |  +--ro clockIdentity?                    clock_identity_type
  |  +--ro portRole?                         port-role-enumeration
  |  +--rw pttPortEnabled?                   boolean
  |  +--ro isMeasuringDelay?                 boolean
  |  +--ro asCapable?                        boolean
  |  +--ro neighborPropDelay
  |  |  +--ro neighborPropDelayHs?   uint32
  |  |  +--ro neighborPropDelayMs?   uint32
  |  |  +--ro neighborPropDelayLs?   uint32
  |  +--rw neighborPropDelayThresh
  |  |  +--rw neighborPropDelayThreshHs?   uint32
  |  |  +--rw neighborPropDelayThreshMs?   uint32
  |  |  +--rw neighborPropDelayThreshLs?   uint32
  |  +--rw delayAsymmetry
  |  |  +--rw delayAsymmetryHs?   uint32
  |  |  +--rw delayAsymmetryMs?   uint32
  |  |  +--rw delayAsymmetryLs?   uint32
  |  +--ro neighborRateRatio?                int32
  |  +--rw initialLogAnnounceInterval?       int32
  |  +--ro currentLogAnnounceInterval?       int32
  |  +--rw announceReceiptTimeout?           uint32
  |  +--rw initialLogSyncInterval?           int32
  |  +--rw currentLogSyncInterval?           int32
  |  +--rw syncReceiptTimeout?               uint32
  |  +--ro syncReceiptTimeoutTimeInterval
  |  |  +--ro syncReceiptTimeoutTimeIntervalHs?   uint32
```

```
  |  |  +--ro syncReceiptTimeoutTimeIntervalMs?   uint32
  |  |  +--ro syncReceiptTimeoutTimeIntervalLs?   uint32
  |  +--rw initialLogPdelayReqInterval?      int32
  |  +--ro currentLogPdelayReqInterval?      int32
  |  +--rw allowedLostResponses?             uint32
  |  +--ro versionNumber?                    uint32
  |  +--rw nup?                              uint64
  |  +--rw ndown?                            uint64
  |  +--rw acceptableMasterTableEnabled?     boolean
  +--ro portStatIfEntry* [portNumber]
  |  +--ro asIfIndex?                                  if-index
  |  +--ro portNumber                                  uint32
  |  +--ro statRxSyncCount?                            yang:counter32
  |  +--ro statRxFollowUpCount?                        yang:counter32
  |  +--ro statRxPdelayRequest?                        yang:counter32
  |  +--ro statRxPdelayResponse?                       yang:counter32
  |  +--ro statRxPdelayResponseFollowUp?               yang:counter32
  |  +--ro statRxAnnounce?                             yang:counter32
  |  +--ro statRxPTPPacketDiscard?                     yang:counter32
  |  +--ro statRxSyncReceiptTimeouts?                  yang:counter32
  |  +--ro statAnnounceReceiptTimeouts?                yang:counter32
  |  +--ro statPdelayAllowedLostResponsesExceeded?     yang:counter32
  |  +--ro statTxSyncCount?                            yang:counter32
  |  +--ro statTxFollowUpCount?                        yang:counter32
  |  +--ro statTxPdelayRequest?                        yang:counter32
  |  +--ro statTxPdelayResponse?                       yang:counter32
  |  +--ro statTxPdelayResponseFollowUp?               yang:counter32
  |  +--ro statTxAnnounce?                             yang:counter32
  +--rw acceptableMasterTableDataSet
     +--ro acceptableMasterTableDSMaxTableSize?            uint16
     +--rw acceptableMasterTableDSActualTableSize?         uint16
     +--rw acceptableMasterTableDSacceptableMasterArray*
[acceptableMasterTableDSMasterId]
        +--rw acceptableMasterTableDSMasterId       uint32
        +--rw acceptableMasterPortIdentity
        |  +--rw clockIdentity?   clock_identity_type
        |  +--rw portNumber?      uint32
        +--rw acceptableMasterAlternatePriority1?   uint32
        +--rw acceptableMasterRowStatus?            uint32
```

Figure 1: IEEE 802.1AS Model Structure

5. IEEE 802.1AS YANG data model

```
<CODE BEGINS>
module ietf-gptp {

   namespace "urn:ietf:params:xml:ns:yang:ietf-gptp";
```

```
   prefix "gptp";

   import ietf-yang-types { prefix "yang";}
   import ieee802-dot1q-bridge { prefix "dot1q";}
   import ietf-interfaces { prefix "if";}

   organization "TBD";
   contact
      "Editor: Siwar Ben Hadj Said
               Email:siwar.benhadjsaid@cea.fr
               Michael Boc
               Email:michael.boc@cea.fr>";

   description
      "This module contains the YANG definition to configure
          IEEE 802.1AS-2011 timing and synchronization in IEEE 802.1Q
      bridged Local Area Networks.";

   revision "2018-03-28" {
      description "version 1.0";
      reference "draft-boc-ietf-gptp-yang-00.txt";
   }

   typedef clock_identity_type {
      type binary {
         length 8;
      }
      description
       "Represents an IEEE 802.1 MAC address represented in the
        'canonical' order defined by IEEE 802.1a, EUI-64. EUI-48
        converts to EUI-64 as specified by IEEE. The conversion
        assigns values 255 and 254 to octets 3 and 4, respectively,
        where octet 0 is the most significant and octet 7 the least.
        For example, EUI-48 of AC:DE:48:23:45:67 would extend to
        AC:DE:48:FF:FE:23:45:67.";
   }

   typedef time-source-enumeration {
      type enumeration {
         enum ATOMIC_CLOCK {
            value 16;
            description
             "Any device, or device directly connected to such a
              device, that is based on atomic resonance for frequency
              and that has been calibrated  against international
              standards for frequency and time";
         }
         enum GPS {
```

```
            value 32;
            description
               "Any device synchronized to any of the satellite
                systems that distribute time and frequency tied to
                international standards";
         }
         enum TERRESTRIAL_RADIO {
            value 48;
            description
            "Any device synchronized via any of the radio distribution
             systems that distribute time and frequency tied to
             international standards";
         }
         enum PTP {
            value 64;
            description
             "Any device synchronized to an IEEE 1588 PTP-based source
              of time external to the gPTP domain.";
         }
         enum NTP {
            value 80;
            description
             "Any device synchronized via NTP to servers that distribute
              time and frequency tied to international standards";
         }
         enum HAND_SET {
            value 96;
            description
             "Used in all cases for any device whose time has been set
              by means of a human interface based on observation of an
              international standards source of time to within the claimed
              clock accuracy";
         }
         enum OTHER {
            value 144;
            description
             "Any source of time and/or frequency not covered by other
              values, or for which the source is not known";
         }
         enum INTERNAL_OSCILLATOR {
            value 160;
            description
             "Any device whose frequency is not based on atomic resonance
              nor calibrated against international standards for frequency,
              and whose time is based on a free-running oscillator with
              epoch determined in an arbitrary or unknown manner";
         }
      }
```

```
      description
       "It indicates the type of source of time used by a clock master.";
      reference
       "IEEE 802.1AS: clause 8.6.2.7 and Table 8-3";
   }

   typedef port-role-enumeration {
      type enumeration {
         enum disabledPort {
            value 3;
            description
             "The port shall not place any messages on its communication
              path.
              In a boundary clock, no activity at the port shall be allowed
              to affect the activity at any other port of the boundary clock.
              A port in this state shall discard all PTP received messages
              except for management messages.
              Any port of the time-aware system for which portEnabled,
              pttPortEnabled, and asCapable are not all TRUE.";
         }
         enum masterPort {
            value 6;
            description
             "The port is behaving as a master port.";
         }
         enum passivePort {
            value 7;
            description
             "The port shall not place any messages on its communication
              path except for Pdelay_Req, Pdelay_Resp,
              Pdelay_Resp_Follow_Up,
              or signaling messages, or management messages that are a
              required response to another management message.";
         }
         enum slavePort {
            value 9;
            description
             "The port is behaving as a slave port.";
         }
      }
      description
       "It indicates the different roles that a port could have";
   }

   typedef if-index {
      type leafref {
       path "/if:interfaces-state/if:interface/if:if-index";
      }
```

```
      description
       "This type is used by the data model to reference network
        interfaces in the system.";
   }

   typedef bridge-port-number {
      type leafref {
      path "/if:interfaces/if:interface/dot1q:bridge-port/dot1q:port-number";
      }
      description
       "An integer that uniquely identifies a Bridge Port.";
   }

   grouping port-identity-group {
      description
       "represents the port identity and consists of two attributes:
        clockIdentity and portNumber";
      leaf clockIdentity {
         type clock_identity_type;
         description
            "Globally unique manufacturer-assigned clock identitifier
             for the localClock.
             The identifier is based on an EUI-64.";
         reference
          "IEEE 802.1AS: clause 8.5.2.2";
      }
      leaf portNumber {
         type uint32;
         default 0;
         description
          "The portNumber value for a port on a time-aware end station
           (i.e., a time-aware system supporting a single gPTP port)
           shall be 1.
           The portNumber values for the gPTP ports on a time-aware
           bridge supporting N ports shall be 1, 2, ..., N";
         reference
          "IEEE 802.1AS: clause 8.5.2.3";
      }
   }

   grouping clock-quality-grouping {
      description
       "describes the clock quality";
      reference
       "IEEE 802.1AS: clause 6.3.3.8";
      leaf clockClass{
         type uint8;
         description
```

```
          "The clockClass denotes the traceability of the time or
           frequency of the clock.
           The value of the clockClass shall be selected as follows:
             a) if the Default Data set member gmCapable is TRUE, then
                clockClass is set to the value that reflects the
                combination of the LocalClock and ClockSource entities;
                else if the value that reflects the LocalClock and
                ClockSource entities is not specified or not known,
                clockClass is set 255.
             b) If the Default Data set member gmCapable is FALSE,
                clockClass is set to 255.";
        reference
         "IEEE 802.1AS: clause 8.6.2.2";
      }
      leaf clockAccuracy {
        type uint8;
        description
         "The clockAccuracy indicates the expected accuracy of the clock.
          It characterize the local clock for the purpose of the best
          master clock algorithm. The value shall be selected as follows:
            a) clockAccuracy is set to the value that reflects the
               combination of the LocalClock and ClockSource entities
               if specified or known;
            b) if the value that reflects the LocalClock and ClockSource
               entities is not specified or unknown, clockAccuracy is set
               to 254.";
        reference
         "IEEE 802.1AS: clause 8.6.2.3";
      }
      leaf offsetScaledLogVariance {
        type uint16;
        description
         "The offsetScaledLogVariance represents an estimate of the PTP
          variance.
          The PTP variance characterizes the precision and frequency
          stability of the ClockMaster. The PTP variance is the square of
          PTP deviation PTPDEV.
          The value shall be selected as follows:
           a) offsetScaledLogVariance is set to the value that reflects the
              combination of the LocalClock and ClockSource entities; else
           b) if the value that reflects these entities is not specified
              or not known, offsetScaledLogVariance is set to 16640
              (410016).
              This value corresponds to the value of PTPDEV for observation
              interval equal to the default Sync message transmission
              interval (i.e., observation interval of 0.125 s).
          A value of -37268 indicates value too large to be represented or
          has been  computed.";
```

```
         reference
          "IEEE 802.1AS: clause 8.6.2.4";
      }
   }

   grouping offset-from-master-group {
      description
       "The value is an implementation-specific representation of the
        current value of the time difference between a slave and the
        grandmaster, as computed by the slave. It is recommended that
        the data type be scaledNs (96-bits integer).
        The default value is implementation specific.";
      leaf offsetFromMasterHs {
         type int32;
         units "2**-16 ns * 2**64";
         description
          "The most significant 32 bits of the offset, signed 96 bit number
           in  2**-16 ns, an implementation-specific computation of the
           current value of the time difference between a master and a slave
           as computed by the slave.
           This parameter MUST be read at the same time as
           offsetFromMasterMs,
           and offsetFromMasterLs, which represents middle and least
           significant 32 bits of values, respectively, in order for
           the read operation to succeed.";
      }
      leaf offsetFromMasterMs {
         type int32;
         units "2**-16 ns * 2**32";
         description
            "The middle significant 32 bits of the offset, signed 96 bit
             number in  2**-16 ns, an implementation-specific computation of
             the current value of the time difference between a master and a
             slave as computed by the slave.
             This parameter MUST be read at the same time as
             offsetFromMasterHs,
             and offsetFromMasterLs,
             which represents most (highest) and
             least significant 32 bits of values, respectively, in order
             for the read  operation to succeed.";
      }
      leaf offsetFromMasterLs {
         type int32;
         units "2**-16 ns";
         description
          "The least significant 32 bits of the offset, signed 96 bit number
           in 2**-16 ns, an implementation-specific computation of the
           current value of  the time difference between a master and
```

```
          a slave as computed by the slave.
          This parameter MUST be read at the same time as
          offsetFromMasterHs,
          and offsetFromMasterMs,
          which represents most (highest) and
          middle significant 32 bits of values, respectively, in order for
          the read operation to succeed.";
     }
   }

   grouping last-gm-phase-change-group {
     description
     "The value is the phase change that occurred on the most recent
      change in either grandmaster or gmTimeBaseIndicator.";
     leaf lastGmPhaseChangeHs {
        type uint32;
        description
         "The value is the phase change that occurred on the most recent
          change in either grandmaster or gmTimeBaseIndicator.
          This parameter MUST be read at the same time as
          lastGmPhaseChangeMs,
          and lastGmPhaseChangeLs,
          which represents middle and least
          significant 32 bits of values, respectively, in order for
          the read operation to succeed.";
     }
     leaf lastGmPhaseChangeMs {
        type uint32;
        description
         "The value is the phase change that occurred on the most recent
          change in either grandmaster or gmTimeBaseIndicator.
          This parameter MUST be read at the same time as
          lastGmPhaseChangeHs,
          and lastGmPhaseChangeLs,
          which represents most and least
          significant 32 bits of values, respectively, in order for
          the read  operation to succeed.";
     }
     leaf lastGmPhaseChangeLs {
        type uint32;
        description
         "The value is the phase change that occurred on the most recent
          change in either grandmaster or gmTimeBaseIndicator.
          This parameter MUST be read at the same time as
          lastGmPhaseChangeMs,
          and lastGmPhaseChangeLs, which represents middle and least
          significant 32 bits of values, respectively, in order for
          the read operation to succeed.";
```

```
    }
  }

  grouping last-gm-freq-change-group {
    description
     "The value is the frequency change that occurred on the most recent
      change in either grandmaster or gmTimeBaseIndicator.";
    leaf lastGmFreqChangeMs {
       type int32;
       description
        "The value is the frequency change that occurred on the
         most recent change in either grandmaster or
         gmTimeBaseIndicator.
         This parameter MUST be read at the same time as
         lastGmFreqChangeLs,
         which represents least significant 32 bits of the value
         in order for the read operation to succeed.";
    }
    leaf lastGmFreqChangeLs {
       type int32;
       description
        "The value is the frequency change that occurred on the
         most recent change in either grandmaster or
         gmTimeBaseIndicator.
         This parameter MUST be read at the same time as
         lastGmFreqChangeMs,
         which represents most significant 32 bits of the value in
         order for the read operation to succeed.";
    }
  }
  grouping neighbor-prop-delay-group {
     description
      "It is an estimate of the current one-way propagation time
       on the link attached to this port, measured as specified for
       the respective medium.
       The value is zero for ports attached to IEEE 802.3 EPON links
       and for the master port of an IEEE 802.11 link, because
       one-way propagation
       delay is not measured on the latter and not directly measured
       on the former. It is recommended that the data type be scaled in ns.
       The initialization value is zero.";
     leaf neighborPropDelayHs {
        type uint32;
        units "2^-16 ns * 2^64";
        description
         "The most (highest) significant 32 bits, unsigned 96 bit number
          in 2**-16 ns, the value is equal to the value of the per-port
          global variable neighborPropDelay.
```

```
          This parameter MUST be read at the same time as
          neighborPropDelayMs, and neighborPropDelayLs,
          which represents middle and least significant 32 bits of values,
          respectively, in order for the read operation to succeed.";
      }
      leaf neighborPropDelayMs {
         type uint32;
         units "2^-16 ns * 2^32";
         description
          "The second most (middle) significant 32 bits, unsigned 96 bit
           number in 2**-16 ns, the value is equal to the value of the
           per-port global variable neighborPropDelay.
           This parameter MUST be read at the same time as
           neighborPropDelayHs,
           and neighborPropDelayLs, which represents most (highest)
           and least significant 32 bits of values, respectively, in
           order for the read operation to succeed.";
      }
      leaf neighborPropDelayLs {
         type uint32;
         units "2^-16 ns";
         description
          "The least significant 32 bits, unsigned 96 bit number in
           2**-16 ns,the value is equal to the value of the per-port
           global variable neighborPropDelay.
           This parameter MUST be read at the same time as
           neighborPropDelayHs,
           and neighborPropDelayMs, which represents most (highest)
           and middle significant 32 bits of values, respectively,
           in order for the read operation to succeed.";
      }
   }
   grouping sync-receipt-timeout-time-interval-group{
      description
       "It is the time interval after which sync receipt timeout
        occurs iftime-synchronization information has not been
        received during the interval.";
      leaf syncReceiptTimeoutTimeIntervalHs {
         type uint32;
         units "2^-16 ns * 2^64";
         description
          "The most (highest) significant 32 bits, of unsigned 96 bit
           number in 2**-16 ns.
           This parameter MUST be read at the same time as
           syncReceiptTimeoutTimeIntervalMs, and
           syncReceiptTimeoutTimeIntervalLs, which represents middle
           and least significant 32 bits of values, respectively, in
           order for the read operation to succeed.
```

```
            Default value is calculated per 10.2.4.2, or
            '0000 0000 0000 165A 0BC0 0000'h.
            The contents of this variable SHALL be maintained across a
            restart of the system.";
      }
      leaf syncReceiptTimeoutTimeIntervalMs {
         type uint32;
         units "2^-16 ns * 2^32";
         description
          "The middle significant 32 bits, unsigned 96 bit number in
           2**-16 ns.
           This parameter MUST be read at the same time as
           syncReceiptTimeoutTimeIntervalHs, and
           syncReceiptTimeoutTimeIntervalLs,
           which represents most (highest) and least significant
           32 bits of values, respectively, in order for the read
           operation to succeed.
           Default value is calculated per 10.2.4.2, or
           '0000 0000 0000 165A 0BC0 0000'h.
           The contents of this variable SHALL be maintained across
           a restart of the system.";
      }
      leaf syncReceiptTimeoutTimeIntervalLs {
         type uint32;
         units "2^-16 ns";
         description
          "The least significant 32 bits, unsigned 96 bit number in
           2**-16 ns.
           This parameter MUST be read at the same time as
           syncReceiptTimeoutTimeIntervalHs, and
           syncReceiptTimeoutTimeIntervalMs,
           which represents most (highest) and middle significant
           32 bits of values, respectively, in order for the read
           operation to succeed.
           Default value is calculated per 10.2.4.2, or
           '0000 0000 0000 165A 0BC0 0000'h.
           The contents of this variable SHALL be maintained across
           a restart of the system.";
      }
   }

   grouping neighbor-prop-delay-thresh-group {
      description
       "It is the propagation time threshold, above which a port is
        not considered capable of participating in the 802.1AS protocol.";
      leaf neighborPropDelayThreshHs {
         type uint32;
         units "2^-16 ns * 2^64";
```

```
         description
          "The most (highest) significant 32 bits of
           neighborPropDelayThresh,
           unsigned 96 bit number in 2**-16 ns.
           This parameter MUST be read or written at the same time as
           neighborPropDelayThreshMs, and
           neighborPropDelayThreshLs, which represents middle and least
           significant 32 bits of values, respectively, in order for the
           read or write operation to succeed.
           The contents of this variable SHALL be maintained across a
           restart of the system.";
      }
      leaf neighborPropDelayThreshMs {
         type uint32;
         units "2^-16 ns * 2^32";
         description
          "The middle significant 32 bits of neighborPropDelayThresh,
           unsigned 96 bit number in 2**-16 ns.
           This parameter MUST be read or written at the same time as
           neighborPropDelayThreshHs, and
           neighborPropDelayThreshLs, which represents most (highest)
           and least significant 32 bits of values, respectively, in
           order for the read or write operation to succeed.
           The contents of this variable SHALL be maintained across a
           restart of the system.";
      }
      leaf neighborPropDelayThreshLs {
         type uint32;
         units "2^-16 ns";
         description
          "The least significant 32 bits of neighborPropDelayThresh,
           unsigned 96 bit number in 2**-16 ns.
           This parameter MUST be read at the same time as
           neighborPropDelayThreshHs,
           and neighborPropDelayThreshMs, which represents most (highest)
           and middle significant 32 bits of values, respectively,
           in order for the read or write operation to succeed.
           The contents of this variable SHALL be maintained across a
           restart of the system.";
      }
   }

   grouping delay-asymmetry-group {
      description
       "The value is the asymmetry in the propagation delay on the
        link attached to this port relative to the grandmaster time base.
        If propagation delay asymmetry is not modeled,
        then delayAsymmetry is 0.";
```

```
      leaf delayAsymmetryHs {
         type uint32;
         units "2^-16 ns * 2^64";
         description
          "The most (highest) significant 32 bits of delayAsymmetry,
           signed 96 bit number in 2**-16 ns.
           This leaf MUST be read or written at the same time as
           delayAsymmetryMs, and
           delayAsymmetryLs, which represents middle and least
           significant 32 bits of values, respectively, in order for
           the read or write operation to succeed.
           The contents of this variable SHALL be maintained across
           a restart of the system.";
      }
      leaf delayAsymmetryMs {
         type uint32;
         units "2^-16 ns * 2^32";
         description
          "The middle significant 32 bits of delayAsymmetry,
           signed 96 bit number in 2**-16 ns.
           This leaf MUST be read or written at the same time as
           delayAsymmetryHs, and
           delayAsymmetryLs, which represents middle and least
           significant 32 bits of values, respectively, in order for
           the read or write operation to succeed.
           The contents of this variable SHALL be maintained across
           a restart of the system.";
      }
      leaf delayAsymmetryLs {
         type uint32;
         units "2^-16 ns";
         description
          "The least significant 32 bits, signed 96 bit number in
           2**-16 ns.
           This object MUST be read or written at the same time as
           delayAsymmetryHs, and
           delayAsymmetryLs, which represents most (highest) and
           least significant 32 bits of values, respectively, in
           order for the read or write operation to succeed.

           The contents of this variable SHALL be maintained across
           a restart of the system.";
      }
   }

   container defaultDataSet {
      description
       "The default data set represents native time capability of a
```

```
        time-aware system and is consistent with respective
        IEEE 1588 data set.";
      leaf clockIdentity {
             type clock_identity_type;
         config false;
         description
          "Globally unique manufacturer-assigned clock identifier
           for the local clock. The identifier is based on an EUI-64.";
         reference
          "IEEE 802.1AS: clause 14.2.1";
      }
      leaf numberPorts {
         type uint32 {
            range "0 .. 255";
         }
         config false;
         description
           "The number of PTP ports on the device.
            For an end station the value is 1.";
         reference
          "IEEE 802.1AS: clause 14.2.2";
      }
      container clockQuality {
             config false;
         description
          "It reflects the quality clock of the LocalClock entity.";
         uses clock-quality-grouping;
      }
      leaf priority1 {
         type uint32 {
            range "0 .. 255";
         }
         default 246;
         description
          "Most-significant priority declaration in the execution
           of the best master clock algorithm. Lower values take
           precedence. The value of priority1 shall be 255 for a
           time-aware system that is not grandmaster-capable.
           The value of priority1 shall be less than 255 for a
           time-aware system that is grandmaster-capable.
           The value 0 shall be reserved for future
           management use, i.e., the value of priority1 shall be set
           to 0 only via management action, and shall not be specified
            as a default value by a user of this standard.
           In the absence of a default value set by a user of this
           standard, the default value shall be set as below:
               a) system type of network infrastructure time-aware
                  system to value 246;
```

```
            b) portable time-aware system, 250;
            c) other time-aware systems, 248.";
         reference
          "IEEE 802.1AS: clause 14.2.7";
      }
      leaf priority2 {
         type uint32 {
            range "0 .. 255";
         }
         default 248;
         description
          "Least-significant priority declaration in the execution
           of the best  master clock algorithm.
           Lower values take precedence.
           The default value is 248.";
         reference
          "IEEE 802.1AS: clause 14.2.7";
      }
      leaf gmCapable {
         type boolean;
         config false;
         description
          "True (1) if master clock capable; false (2)otherwise";
         reference
          "IEEE 802.1AS: clause 14.2.8";
      }
      leaf currentUtcOffset {
         type uint32;
         description
          "The value is the offset between TAI and UTC, relative
           to the ClockMaster entity of this time-aware system.
           It is equal to the global variable sysCurrentUtcOffset.
           The value is in units of seconds.
           The initialization default value is selected as follows:
              a) the value is the value obtained from a primary
                 reference if the  value is known at the time of
                 initialization,
              b) else the value is the current number of leap
                 seconds, when the time-aware system is designed.";
         reference
          "IEEE 802.1AS: clause 14.2.9";
      }
      leaf currentUtcOffsetValid {
         type boolean;
         description
          "True (1) if currentUTCOffset is known to be correct;
           false (2) otherwise.";
         reference
```

```
        "10.2.10";
      }
      leaf leap59 {
         type boolean;
         description
          "A true (1) value indicates that the last minute of the
           current UTC day, relative to the ClockMaster entity of
           this time-aware system, will contain 59 seconds.
           It is equal to the global variable sysLeap59.
           The initialization value is selected as follows:
             a) Set to true (1) if the value is obtained from a
                primary reference if known at the at the time of
                initialization, else
             b) The value is set to false (2).";
         reference
          "IEEE 802.1AS: clause 14.2.11";
      }
      leaf leap61 {
         type boolean;
         description
          "A true (1) value indicates that the last minute of the
           current UTC day, relative to the ClockMaster entity of
           this time-aware system, will contain 59 seconds.
           It is equal to the global variable sysLeap61.
           The initialization value is selected as follows:
               a) Set to true (1) if the value is obtained from a
                  primary reference if known at the at the time
                  of initialization, else
               b) The value is set to false (2).";
         reference
          "IEEE 802.1AS: clause 14.2.12";
      }
      leaf timeTraceable {
         type boolean;
         config false;
         description
          "The value is set to true (1) if the timescale and the
           value of currentUtcOffset, relative to the ClockMaster
           entity of this time-aware system, are traceable to a
           primary reference standard; otherwise
           the value is set to false (2).
           The initialization value is selected as follows:
              a) If the time and the value of currentUtcOffset are
                 traceable to a primary reference standard at the
                 time of initialization, the value is set to
                 true (1), else
              b) The value is set to false (2).";
         reference
```

```
          "IEEE 802.1AS: clause 14.2.13";
      }
      leaf frequencyTraceable {
         type boolean;
         config false;
         description
          "The value is set to true (1) if the frequency determining
           the timescale of the ClockMaster Entity of this time aware
           system is traceable to a primary reference standard;
           otherwise the value is set to false (2).
           The initialization value is selected as follows:
              a) If the frequency is traceable to a primary reference
                 standard at the time of initialization,
                 the value is set to true (1), else
              b) The value is set to false (2).";
         reference
          "IEEE 802.1AS: clause 14.2.14";
      }
      leaf timeSource {
         type time-source-enumeration;
         default INTERNAL_OSCILLATOR;
         config false;
         description
          "The timeSource is an information-only attribute indicating
           the type of source of time used by a ClockMaster.
           The value is not used in the selection
           of the grandmaster.
           The initialization value is selected as follows:
              a) If the timeSource, is known at the time of
                 initialization, else
              b) The value is set to INTERNAL_OSCILLATOR (160).";
         reference
          "IEEE 802.1AS: clause 14.2.14";
      }
   }

   container currentDataSet {
      config false;
      description
       "The current data set represent this system's topological
        location relative to the known grandmaster system.
                This data set is consistent with respective IEEE 1588
        data set.";
      leaf stepsRemoved {
         type int32;
         description
          "The number of communication paths traversed between the
           local clock and the grandmaster clock. For example,
```

```
            stepsRemoved for a slave clock on the same PTP communication
            path as the grandmaster clock will have a value of 1,
            indicating that a single path was traversed.";
          reference
           "IEEE 802.1AS clause 14.3.1";
        }
        container offsetFromMaster {
          description
           "It represents the current value of the time difference between
            a slave and the grandmaster, as computed by the slave.
            It is recommended that the data type be scaledNs.";
          uses offset-from-master-group;
          reference
           "IEEE 802.1AS clause 14.3.2";
        }
        container lastGmPhaseChange {
          description
           "The value is the phase change that occurred on the most
            recent change in either grandmaster or gmTimeBaseIndicator.";
          uses last-gm-phase-change-group;
          reference
           "IEEE 802.1AS: clause 14.3.3";
        }
        container lastGmFreqChange {
          uses last-gm-freq-change-group;
          description
           "The value is the frequency change that occurred on the most
            recent change in either grandmaster or gmTimeBaseIndicator.";
          reference
           "IEEE 802.1AS: clause 14.3.4";
        }
        leaf gmTimebaseIndicator {
          type uint32;
          description
           "This reports the grandmaster time base change value conveyed
            in the Sync message.
            The value is the value of timeBaseIndicator of the current
            grandmaster";
          reference
           "IEEE 802.1AS: clause 14.3.5";
        }
        leaf gmChangeCount {
          type yang:counter32;
          description
           "This statistics counter tracks the number of times the
            grandmaster has changed in a gPTP domain.";
          reference
           "IEEE 802.1AS: clause 14.3.6";
```

```
      }
      leaf timeOfLastGmChangeEvent {
         type yang:timestamp;
         units "0.01 seconds";
         description
          "This timestamp denotes the system time when the most
           recent grandmaster change occurred in a gPTP domain.";
      }
      leaf timeOfLastGmFreqChangeEvent {
         type yang:timestamp;
         units "0.01 seconds";
         description
          "This timestamp denotes the system time when the most
           recent change in grandmaster phase occured, due to a
           change of either the grandmaster or the grandmaster
           time base.";
      }
      leaf timeOfLastGmPhaseChangeEvent {
         type yang:timestamp;
         units "0.01 seconds";
         description
          "This timestamp denotes the system time when the most recent
           change in grandmaster frequency occured, due to a change of
           either the grandmaster or the grandmaster time base.";
         reference
         "IEEE 802.1AS: clause ";
      }
   }

   container parentDataSet {
      config false;
      description
       "The paraent data set represent timing upstream
           (toward grandmaster) system's parameters as measured
       at this system.
       This data set is consistent with respective IEEE 1588 data set.";
      container parentPortIdentity {
         uses port-identity-group;
         description
           "Clock identifier (clockIdentity) of the parent LocalClock.
           The default value is set to
           /gptp:defaultDataSet/gptp:ClockIdentity.
           If this time-aware system is the grandmaster, the value
           is the clockIdentity of this time-aware system.
           If this time-aware system is not the grandmaster,
           the value is the clockIdentity of the MasterPort of the
           gPTP communication path attached to the single slave port
           of this time-aware system.
```

```
            Port number (portNumber) of the local clock parent gPTP
            port number.
            If this time-aware system is the grandmaster, the value
            is the gPTP portNumber of this time-aware system.
            If this time-aware system is not the grandmaster,
            the value is the portNumber of the MasterPort of the
            gPTP communication path attached to the single gPTP slave
            port of this time-aware system.";
         reference
          "IEEE 802.1AS: clause 14.4.1";

      }

      leaf cumulativeRateRatio {
         type int32;
         description
          "The value is an estimate of the ratio of the frequency of
           the grandmaster to the frequency of the LocalClock entity
           of this time-aware system.
           Cumulative rate ratio is expressed as the fractional
           frequency offset multiplied by 2**41, i.e.,
           the quantity (rateRatio - 1.0)(2**41), where
           rateRatio is computed by the PortSyncSyncReceive
           state machine";
         reference
          "IEEE 802.1AS: clause 14.4.2";
      }
      leaf grandmasterIdentity {
         type clock_identity_type;
         description
          "Clock identifier (clockIdentity) of the grandmaster.
           The default value is set to clockIdentity of the
           defaultDataSet";
         reference
          "IEEE 802.1AS: clause 14.4.3";
      }
      container grandmasterClockQuality {
         uses clock-quality-grouping;
         description
          "Denotes the clock quality of the grandmaster.
           The default value for clockClass is set to defaultDataSet
           ClockClass.
           The default value for clockAccuracy is set to DefaultDataSet
           ClockAccuracy.
           The default value for offsetScaledLogVariance is set to
           DefaultDataSet OffsetScaledLogVariance";
         reference
          "IEEE 802.1AS: clause 14.4.4, 14.4.5 and 14.4.6";
```

```
      }
      leaf grandmasterPriority1 {
         type uint32 {
            range "0 .. 255";
         }
         description
          "Grandmaster's most-significant priority declaration in the
           execution of the best master clock algorithm.
           Lower values take precedence.
           The default value is set to default Data Set Priority1";
         reference
          "IEEE 802.1AS: clause 14.4.7";
      }
      leaf grandmasterPriority2 {
         type uint32 {
            range "0 .. 255";
         }
         description
          "Grandmaster's least-significant priority declaration in
           the execution of the best master clock algorithm.
           Lower values take precedence.
           The default value is set to default Data Set Priority2";
         reference
          "IEEE 802.1AS: clause 14.4.8";
      }
   }

   container timePropertiesDataSet {
      config false;
      description
       "The time properties data set represents the grandmaster's
        parameters, as measured at this system and are derived from
        IEEE 802.1AS protocol.
        This data set is consistent with respective IEEE 1588 data set.";
      leaf currentUTCOffset {
         type int32;
         units "seconds";
         description
          "The value is currentUtcOffset for the current grandmaster.
           The value is in units of
           seconds. The default value is set to currentUTCOffset.";
         reference
          "IEEE 802.1AS: clause 14.5.1";
      }
      leaf currentUTCOffsetValid {
         type boolean;
         description
          "True (1) if currentUTCOffset is known to be correct;
```

```
            false (2) otherwise.
            The default value is set to currentUTCOffsetValid.";
         reference
          "IEEE 802.1AS: clause 14.5.2";
      }
      leaf leap59 {
         type boolean;
         description
          "The value is leap59 for the current grandmaster. It is equal
           to the global variable leap59.
           A true (1) value indicates that the last minute of the
           current UTC day, relative to the ClockMaster entity of this
           time-aware system, will contain 59 seconds.";
         reference
          "IEEE 802.1AS: clause 14.5.3";
      }
      leaf leap61 {
         type boolean;
         description
          "The value is leap61 for the current grandmaster. It is equal
           to the global variable leap61.
           A true (1) value indicates that the last minute of the
           current UTC day, relative to the ClockMaster entity of this
           time-aware system, will contain 61 seconds.";
         reference
          "IEEE 802.1AS: clause 14.5.4";
      }
      leaf timeTraceable {
         type boolean;
         description
          "The value is timeTraceable for the current grandmaster. It
           is equal to the global variable timeTraceable.
           True (1) if the timescale and the value of this parameter
           are traceable to a primary reference; false (2) otherwise.";
         reference
          "IEEE 802.1AS: clause ";
      }
      leaf frequencyTraceable {
         type boolean;
         description
          "The value is frequencyTraceable for the current grandmaster.
           It is equal to the global variable frequencyTraceable.
           True (1) if the frequency determining the timescale is traceable
           to a primary reference; false (2) otherwise.";
         reference
          "IEEE 802.1AS: clause 14.5.6";
      }
      leaf timeSource {
```

```
         type time-source-enumeration;
         description
          "The value is timeSource for the current grandmaster.
           It indicates the source of time used by the grandmaster clock.";
         reference
          "IEEE 802.1AS: clause 14.5.7";
      }
   }

   list portDataSet {
      key "portNumber";
      description
       "A list of gPTP port related variables in a time-aware Bridge or
        for a time-aware end station. A value of 1 is used in a bridge
        or an end station that does not have multiple components.
        For a given media port of a Bridge or an end station, there may
        be one or more gPTP port, and depends whether a media port supports
        point to point link (e.g. IEEE 802.3 Ethernet) or point to
        multi-point (e.g. CSN, IEEE 802.3 EPON,etc) links on the
        media port.";
      leaf bridgeBasePort {
         type bridge-port-number;
         config false;
         mandatory true;
         description
          "It identifies the bridge port number of the port for which this
           entry contains bridge management information.
           For end stations, this port number shall be (1).";
      }
      leaf asIfIndex {
         type if-index;
         config false;
         mandatory true;
         description
          "It identifies the gPTP interface group within the system for
           which this portDataSet contains information.
           It is the value of the instance of the IfIndex object,
           defined in the IF-MIB, for the gPTP interface group corresponding
           to this port, or the value 0 if the port has not been bound
           to an underlying frame source and sink.
           For a given media port of a Bridge or an end station, there may
           be one or more gPTP port, and depends whether a media port
           supports point to point link (e.g. IEEE 802.3 Ethernet) or
           point to multi-point (e.g. CSN, IEEE 802.3 EPON, etc) links
           on the media port.";
      }
      leaf portNumber {
         type uint32;
```

```
         mandatory true;
         description
          "The portNumber value for a port on a time-aware end station (i.e.,
           a time-aware system supporting a single gPTP port) shall be 1.
           The portNumber values for the gPTP ports on a time-aware Bridge
           supporting N ports shall be 1, 2, ..., N, respectively";
         reference
          "IEEE 802.1AS: clause 14.6.2";
      }
      leaf clockIdentity {
         type clock_identity_type;
         config false;
         description
          "The clockIdentity is an 8 octet array formed by mapping an IEEE
           EUI-48 assigned to the time-aware system to IEEE EUI-64 format
           (i.e., to an array of 8 octets).
           The EUI-48 shall be an Ethernet MAC address owned by the
           organization creating the instance of a clockIdentity.
           The organization owning the MAC address shall ensure that the MAC
           address is used in generating only a single instance of a
           clockIdentity, for example by requiring that the MAC address be
           a MAC address embedded in the device identified by the
           clockIdentity.
           The mapping rules for constructing the EUI-64 from the EUI-48
           shall be those specified by the IEEE [B2]. The 8 octets of the
           created IEEE EUI-64 shall be assigned in order to the 8 octet
           array clockIdentity with most significant octet of the
           IEEE EUI-64 assigned to the clockIdentity octet array
           member with index 0.";
         reference
          "IEEE 802.1AS: clause 14.6.2";
      }
      leaf portRole {
         type port-role-enumeration;
         default disabledPort;
         config false;
         description
          "The value is the value of the port role of this port.
           The enumeration values are consistent with IEEE Std 1588-2008.
           The default value is 3 (DisabledPort)";
         reference
          "IEEE 802.1AS: clause 14.6.3";
      }

      leaf pttPortEnabled {
         type boolean;
         description
          "802.1AS function enable for a given port.
```

```
            True (1) if the time-synchronization and best master selection
            functions of the port are enabled;
            False (2) otherwise.
            The contents of this table SHALL be maintained across a
            restart of the system.";
         reference
          "IEEE 802.1AS: clause 14.6.4";
      }

      leaf isMeasuringDelay {
         type boolean;
         config false;
         description
          "True (1) if the port is measuring link propagation delay;
           The value is equal to the value of the Boolean
           isMeasuringPdDelay.
           False (2) otherwise.";
         reference
          "IEEE 802.1AS: clause 14.6.5";
      }

      leaf asCapable {
         type boolean;
         config false;
         description
          "True (1) if and only if it is determined that this time-aware
           system and the time-aware system at the other ends of the
           link attached to this port can interoperate with each other
           via the IEEE 802.1AS protocol;
           False (2) otherwise";
         reference
          "IEEE 802.1AS: clause 14.6.6";
      }

      container neighborPropDelay {
         description
          "It is an
           estimate of the current one-way propagation time on the link
           attached to this port, measured as specified for the
           respective medium.
           The value is zero for ports attached to IEEE 802.3 EPON links
           and for the master port of an IEEE 802.11 link, because
           one-way propagation delay is not measured on the latter and
           not directly measured on the former.
           It is recommended that the data type be scaled in ns.
           The initialization value is zero.";
         uses neighbor-prop-delay-group;
         reference
```

```
          "IEEE 802.1AS: clause 14.6.7";
      }
      container neighborPropDelayThresh {
         description
          "It is the propagation time threshold,
           above which a port is not considered capable of participating
           in the IEEE 802.1AS protocol.";
         uses neighbor-prop-delay-thresh-group;
         reference
          "IEEE 802.1AS: clause 14.6.9";
      }
      container delayAsymmetry {
         description
          "The value is the asymmetry in the propagation delay on the link
           attached to this port relative to the grandmaster time base.
           If the propagation delay asymmetry is not modeled,
           then delayAsymmetry is 0";
         uses delay-asymmetry-group;
         reference
          "IEEE 802.1AS: clause 14.6.9";
      }
      leaf neighborRateRatio {
         type int32;
         config false;
         description
          "The value is an estimate of the ratio of the frequency of the
           LocalClock entity of the time-aware system at the other end of
           the link attached to this port, to the frequency of the
           LocalClock entity of this time-aware system.
           Neighbor rate ratio is expressed as the fractional
           frequency offset multiplied by 2**41, i.e.,
           the quantity (neighborRateRatio - 1.0)(2**41)";
         reference
          "IEEE 802.1AS: clause 14.6.10";
      }
      leaf initialLogAnnounceInterval {
         type int32 {
            range "-128 .. 127";
         }
         default 0;
         description
          "The value is the logarithm to the base 2 of the of the announce
           interval used when
            (a) the port is initialized, or
            (b) a message interval request TLV is received with
                announceInterval field set to 126
           The default value is 0.
           The contents of this variable SHALL be maintained across
```

```
          a restart of the system.";
        reference
         "IEEE 802.1AS: clause 14.6.11";
      }
      leaf currentLogAnnounceInterval {
        type int32 {
          range "-128 .. 127";
        }
        config false;
        description
         "The value is the logarithm to the base 2 of the of the
          current announce transmission interval.
          The currentLogAnnounceInterval specifies the current value
          of the announce interval.
          Every port supports the value 127; the port does not send
          Announce messages when currentLogAnnounceInterval has this value.
          A port may support other values, except for the reserved values
          -128 through -125, inclusive, and 124 through 126, inclusive.
          A port ignores requests for unsupported values.";
        reference
         "IEEE 802.1AS: clause 14.6.12";
      }
      leaf announceReceiptTimeout {
        type uint32 {
          range "0 .. 255";
        }
        default 2;
        description
         "The value of this attribute tells a slave port the number of sync
          intervals to wait without receiving synchronization information,
          before assuming that the master is no longer transmitting
          synchronization information, and that the BMC algorithm needs to
          be run, if appropriate.
          The condition of the slave port not receiving synchronization
          information for syncReceiptTimeout sync intervals is referred to
          as 'sync receipt timeout'.
          The default value is 2.";
        reference
         "IEEE 802.1AS: clause 14.6.13";
      }
      leaf initialLogSyncInterval {
        type int32 {
          range "-128 .. 127";
        }
        default -3;
        description
         "The value is the logarithm to the base 2 of the sync interval
          used when,
```

```
            (a) the port is initialized, or
            (b) a message interval request TLV is received with the
                timeSyncInterval field set to 126.
          The initialization value is -3.
          The contents of this variable SHALL be maintained across a
          restart of the system.";
       reference
        "IEEE 802.1AS: clause 14.6.14";
    }
    leaf currentLogSyncInterval {
       type int32 {
          range "-128 .. 127";
       }
       description
        "The value is the logarithm to the base 2 of the current
         time-synchronization transmission interval";
       reference
        "IEEE 802.1AS: clause 14.6.15";
    }
    leaf syncReceiptTimeout {
       type uint32 {
          range "0 .. 255";
       }
       default 3;
       description
        "The value is the number of time-synchronization transmission
         intervals that a slave port waits without receiving
         synchronization information, before assuming that the master
         is no longer transmitting  synchronization information and that
         the BMCA needs to be run, if appropriate.
         The initialization value is 3.
         The contents of this variable SHALL be maintained across a
         restart of the system.";
       reference
        "IEEE 802.1AS: clause 14.6.16";
    }
    container syncReceiptTimeoutTimeInterval {
       description
        "It is the time interval after which sync receipt timeout occurs
         if time-synchronization information has not been received during
         the interval.";
       uses sync-receipt-timeout-time-interval-group;
       reference
        "IEEE 802.1AS: clause 14.6.17";
    }
    leaf initialLogPdelayReqInterval {
       type int32 {
          range "-128 .. 127";
```

```
        }
        default 0;
        description
         "For full-duplex, IEEE 802.3 media and CSN media that use the peer
          delay mechanism to measure path delay, the value is the logarithm
          to the base 2 of the Pdelay_Req message transmission interval
          used when,
            (a) the port is initialized, or
            (b) a message interval request TLV is received with the
                linkDelayInterval field set to 126.
          For these media, the initialization value is 0.
          For all other media, the value is 127.
          The contents of this variable SHALL be maintained across a
          restart of the system.";
        reference
         "IEEE 802.1AS: clause 14.6.18";
      }
      leaf currentLogPdelayReqInterval {
        type int32 {
           range "-128 .. 127";
        }
        config false;
        description
         "For full-duplex, IEEE 802.3 media and CSN media that use the peer
          delay mechanism
          to measure path delay, the value is the logarithm to the base 2
          of the current Pdelay_Req message transmission interval.
          For all other media, the value is 127.
          The contents of this variable SHALL be maintained across a
          restart of the system.";
        reference
         "IEEE 802.1AS: clause 14.6.19";
      }
      leaf allowedLostResponses {
        type uint32 {
           range "0 .. max";
        }
        default 3;
        description
         "The value is equal to the value of the per-port global variable
          allowedLostResponses.
          It is the number of Pdelay_Req messages for which a valid
          response is not received, above which a port is considered to
          not be exchanging peer delay messages";
        reference
         "IEEE 802.1AS: clause 14.6.20";
      }
      leaf versionNumber {
```

```
         type uint32 {
            range "0 .. 63";
         }
         default 2;
         config false;
         description
          "Indicates the PTP version in use on the port.
           The version number for this standard is set to the value 2.
           The contents of this variable SHALL be maintained across a
           restart of the system.";
         reference
          "IEEE 802.1AS: clause 14.6.21";
      }
      leaf nup {
         type uint64 {
            range "0 .. 2";
         }
         description
          "For an OLT port of an IEEE 802.3 EPON link, the value is the
           effective index of refraction for the EPON upstream wavelength
           light of the optical path.
           The default value is 1.46770 for 1 Gb/s upstream links,
           and 1.46773 for 10 Gb/s upstream links.
           For all other ports, the value is 0.";
         reference
          "IEEE 802.1AS: clause 14.6.22";
      }
      leaf ndown {
         type uint64 {
            range "0 .. 2";
         }
         description
          "For an OLT port of an IEEE 802.3 EPON link, the value is the
           effective index of refraction for the EPON downstream wavelength
           light of the optical path.
           The default value is 1.46805 for 1 Gb/s downstream links,
           and 1.46851 for 10 Gb/s downstream links.
           For all other ports, the value is 0.";
         reference
          "IEEE 802.1AS: clause 14.6.23";
      }
      leaf acceptableMasterTableEnabled {
         type boolean;
         default false;
         description
          "True (1) if acceptableMasterTableEnabled is true and an ONU port
           attached  to an IEEE 802.3 EPON link in a time-aware system.
           False (2), otherwise.
```

```
          The default value is FALSE.";
        reference
         "IEEE 802.1AS: clause 14.6.24";
      }
   }

   list portStatIfEntry {
      key "portNumber";
      config false;
      description
       "A list of time-aware port related counters in a gPTP domain.
        A value 1 is used in a  bridge or an end station that does
        not have multiple components.";
      leaf asIfIndex {
         type if-index;
                config false;
         description
          "It identifies the gPTP interface group within the system for
           which this portDataSet contains information.
           It is the value of the instance of the
           IfIndex object, defined in the IF-MIB, for the gPTP interface
           group corresponding to this port, or the value 0 if the port
           has not been bound to an underlying frame source and sink.
           For a given media port of a Bridge or an end station,
           there may be one or more gPTP port, and depends whether a
           media port supports point to point link (e.g. IEEE 802.3
           Ethernet) or point to multi-point (e.g. CSN, IEEE802.3
           EPON, etc) links on the media port.";
      }
      leaf portNumber {
         type uint32;
         config false;
         description
          "The portNumber value for a port on a time-aware end station
           (i.e., a time-aware system supporting a single gPTP port) shall
           be 1.
           The portNumber values for the gPTP ports on a time-aware Bridge
           supporting N ports shall be 1, 2, ..., N, respectively.";
         reference
          "IEEE 802.1AS: clause 14.6.2";
      }
      leaf statRxSyncCount {
         type yang:counter32;
         config false;
         description
          "A counter that increments every time synchronization information
           is received.";
         reference
```

```
          "IEEE 802.1AS: clause 14.7.2";
      }
      leaf statRxFollowUpCount {
         type yang:counter32;
         config false;
         description
          "A counter that increments every time a Follow_Up message
           is received.";
         reference
          "IEEE 802.1AS: clause 14.7.3";
      }
      leaf statRxPdelayRequest {
         type yang:counter32;
         config false;
         description
          "A counter that increments every time a Pdelay_Req message
           is received";
         reference
          "IEEE 802.1AS: clause 14.7.4";
      }
      leaf statRxPdelayResponse {
         type yang:counter32;
         config false;
         description
          "A counter that increments every time a Pdelay_Resp
           message is received.";
         reference
          "IEEE 802.1AS: clause 14.7.5";
      }
      leaf statRxPdelayResponseFollowUp {
         type yang:counter32;
         config false;
         description
          "A counter that increments every time a
           Pdelay_Resp_Follow_Up message is received.";
         reference
          "IEEE 802.1AS: clause 14.7.6";
      }
      leaf statRxAnnounce {
         type yang:counter32;
         config false;
         description
          "A counter that increments every time an Announce
           message is received.";
         reference
          "IEEE 802.1AS: clause 14.7.7";
      }
      leaf statRxPTPPacketDiscard {
```

```
         type yang:counter32;
         config false;
         description
          "A counter that increments every time a PTP message
           is discarded, caused by the occurrence of any of
           the following conditions:
              a) A received Announce message is not qualified.
              b) A Follow_Up message corresponding to a received Sync
                 message is not received.
              c) A Pdelay_Resp message corresponding to a transmitted
                 Pdelay_Req message is not received.
              d) A Pdelay_Resp_Follow_Up message corresponding to a
                 transmitted Pdelay_Req message is not received.";
         reference
          "IEEE 802.1AS: clause 14.7.8";
      }
      leaf statRxSyncReceiptTimeouts {
         type yang:counter32;
         config false;
         description
          "A counter that increments every time sync receipt
           timeout occurs.";
         reference
          "IEEE 802.1AS: clause 14.7.9";
      }
      leaf statAnnounceReceiptTimeouts {
         type yang:counter32;
         config false;
         description
          "A counter that increments every time announce receipt
           timeout occurs.";
         reference
          "IEEE 802.1AS: clause 14.7.10";
      }
      leaf statPdelayAllowedLostResponsesExceeded {
         type yang:counter32;
         config false;
         description
          "A counter that increments everytime the value of the
           variable lostResponses exceeds the value of the
           variable allowedLostResponses.";
         reference
          "IEEE 802.1AS: clause 14.7.11";
      }
      leaf statTxSyncCount {
         type yang:counter32;
         config false;
         description
```

```
          "A counter that increments every time synchronization
           information is transmitted.";
        reference
         "IEEE 802.1AS: clause 14.7.12";
      }
      leaf statTxFollowUpCount {
        type yang:counter32;
        config false;
        description
         "A counter that increments every time a Follow_Up
          message is transmitted.";
        reference
         "IEEE 802.1AS: clause 14.7.13";
      }
      leaf statTxPdelayRequest {
        type yang:counter32;
        config false;
        description
         "A counter that increments every time a Pdelay_Req
          message is transmitted.";
        reference
         "IEEE 802.1AS: clause 14.7.14";
      }
      leaf statTxPdelayResponse {
        type yang:counter32;
        config false;
        description
         "A counter that increments every time a Pdelay_Resp
          message is transmitted.";
        reference
         "IEEE 802.1AS: clause 14.7.15";
      }
      leaf statTxPdelayResponseFollowUp {
        type yang:counter32;
        config false;
        description
         "A counter that increments every time a
          Pdelay_Resp_Follow_Up message is
          transmitted.";
        reference
         "IEEE 802.1AS: clause 14.7.16";
      }
      leaf statTxAnnounce {
        type yang:counter32;
        config false;
        description
         "A counter that increments every time an
          Announce message is transmitted.";
```

```
        reference
         "IEEE 802.1AS: clause 14.7.17";
      }
   }
   container acceptableMasterTableDataSet {
      description
       "Acceptable Master Table derived from IEEE 1588-2008.
        One Table per time-aware system, and used when any
        of the system is of type IEEE 802.3 EPON,
        i.e. if any port in a corresponding system has
        acceptableMasterTableEnabled set to true.
        Not used otherwise (Table exists without an entry).";
      leaf acceptableMasterTableDSMaxTableSize {
         type uint16;
         config false;
         description
          "The value is the maximum size of the AcceptableMasterTable.";
         reference
          "IEEE 802.1AS: clause 14.8.1 and 13.1.3.2";
      }
      leaf acceptableMasterTableDSActualTableSize {
         type uint16;
         description
          "The value is the actual size of the AcceptableMasterTable,
           i.e., the current number of elements in the acceptable
           master array.
           The actual table size is less than or equal to the max
           table size.
           For a time-aware system that contains an ONU attached
           to an IEEE 802.3 EPON link, the initialization value is 1.
           For a time-aware system that does not contain an ONU
           attached to an IEEE 802.3 EPON link,
           the initialization value is 0";
         reference
          "IEEE 802.1AS: clause 14.8.2 and 13.1.3.2";
      }
      list acceptableMasterTableDSacceptableMasterArray {
         key "acceptableMasterTableDSMasterId";
         description
          "A list of time-aware port related variables in a
           time-aware bridge or for a time-aware end station.
           A value of 1 is used in
           a bridge or an end station that does not have
           multiple components.";

         leaf acceptableMasterTableDSMasterId {
            type uint32;
            description
```

```
             "Acceptable Master row entry index in this
              acceptableMasterTabdDSMaster Entry applies.
              If the does not contain Media type of EPON,
              this variable (index) MUST be equal to 0.";
            reference
             "IEEE 802.1AS: clause 14.8.3";
          }
          container acceptableMasterPortIdentity {
            description
             "It represents the identity of a port or aggregate port
              on a bridge component or end-station.";
            uses port-identity-group;
            reference
             "IEEE 802.1AS: clause 14.8.3";
          }
          leaf acceptableMasterAlternatePriority1 {
            type uint32;
            description
             "If the alternatePriority1 member of the AcceptableMaster
              array element that corresponds to the sourcePortIdentity
              of a received Announce message is greater than 0, the
              value of the grandmasterPriority1 field of the Announce
              message is replaced by the value of alternatePriority1
              of this AcceptableMaster array element for use in the
              invocation of BMCA.";
            reference
             "IEEE 802.1AS: clause 14.8.3";
          }
          leaf acceptableMasterRowStatus {
            type uint32;
            description
             "It indicates the status of an entry, and is used to
              create/delete entries.";
            reference
             "IEEE 802.1AS: clause 14.8.3";
          }
        }
    }
}
<CODE ENDS>
```

6. Security Considerations

TBD.

7. IANA Considerations

IANA?

8. Contributors

if needed.

9. Acknowledgements

10. References

10.1. Normative References

[I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-04 (work in progress), October 2017.

[I-D.ietf-detnet-flow-information-model]
Farkas, J., Varga, B., rodney.cummings@ni.com, r., Jiang, Y., and Y. Zha, "DetNet Flow Information Model", draft-ietf-detnet-flow-information-model-01 (work in progress), March 2018.

[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>.

[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>.

[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>.

10.2. Informative References

[I-D.draft-ietf-netconf-restconf]
Bierman, L., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", I-D draft-ietf-netconf-restconf-13, April 2016.

[I-D.ietf-detnet-use-cases]
Grossman, E., "Deterministic Networking Use Cases", draft-ietf-detnet-use-cases-14 (work in progress), February 2018.

[IEEE8021AS]
"IEEE 802.1, "IEEE 802.1AS-2011: IEEE Standard for Local and metropolitan area networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks", 2011, <http://ieeexplore.ieee.org/document/5741898/>.", 2011.

[IEEE8021Qcc]
"IEEE, "Stream Reservation Protocol (SRP) Enhancements and Performance Improvements (IEEE Draft P802.1Qcc)", 2017, <http://www.ieee802.org/1/files/private/cc-drafts/>.".

[IEEE8021TSN]
IEEE 802.1, "IEEE 802.1 Time-Sensitive Networking (TSN) Task Group", <http://www.ieee802.org/1/pages/tsn.html>.

Authors' Addresses

Siwar Ben Hadj Said
CEA, LIST
CEA Saclay
Gif-sur-Yvette , Ile-de-France 91190
France

Phone: +33169082939
Email: Siwar.BenHadjSaid@cea.fr

Michael Boc
CEA, LIST
CEA Saclay
Gif-sur-Yvette , Ile-de-France 91190
France

Phone: +33169089223
Email: Michael.Boc@cea.fr