

Internet Working Group

Y. Jiang

Internet Draft

X. Liu

J. Xu

Huawei

Intended status: Standards Track

Expires: January 2015

July 6, 2015

**Yang Data Model for IEEE 1588v2
draft-jlx-tictoc-1588v2-yang-00.txt**

Status of this Memo

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

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

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

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

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

This Internet-Draft will expire on January 6, 2013.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in

Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Simplified BSD License.

Abstract

This document defines a YANG data model for the configuration of IEEE 1588v2 devices and clocks, and also retrieval of the configuration information, data set and running states of IEEE 1588v2 clocks.

Table of Contents

1.	Introduction	2
2.	Conventions used in this document	3
3.	Terminology	3
4.	IEEE 1588V2 YANG Model	4
4.1.	IEEE 1588V2 YANG model hierarchy	4
4.2.	IEEE 1588v2 rpc definitions	7
5.	IEEE 1588v2 YANG Module	12
6.	Security Considerations	36
7.	IANA Considerations	36
8.	References	36
8.1.	Normative References	36
8.2.	Informative References	36
9.	Acknowledgments	37

[1. Introduction](#)

IEEE 1588v2 as a synchronization protocol is widely supported in the carrier networks. It can provide high precision time synchronization as high as nano-seconds. The protocol depends on a PTP engine to automatically decide its state, and a PTP transportation layer to carry the PTP timing and various quality messages. The configuration parameters and state data sets of IEEE 1588v2 are numerous.

Some work on IEEE 1588v2 MIB [[PTP-MIB](#)] is in progress in the IETF TICTOC WG. But the work is only scoped with retrieval of the state data of IEEE 1588v2 by SNMP and configuration is not considered, thus its use is limited.

Some service providers require the management of the IEEE 1588v2 synchronization network can be more flexible and more Internet-based (typically overlaid on their transport networks). SDN is another driving factor which demands a greater control over

synchronization networks. Advantages of YANG include Internet based configuration capability, validation, roll-back and etc., all these characteristics make it attractive to become a modeling language for IEEE 1588v2.

YANG [[RFC6020](#)] is a data modeling language used to model configuration and state data manipulated by the Network Configuration Protocol (NETCONF) [[RFC6241](#)]. A small set of built-in data types are defined in [[RFC6020](#)], and a collection of common data types are further defined in [[RFC6991](#)].

This document defines a YANG [[RFC6020](#)] data model for the configuration of IEEE 1588v2 devices and clocks, and also retrieval of the state data of IEEE 1588v2 clocks.

In order to fulfill the need of a lightweight implementation, the core module is designed to be generic and minimal, but be extensible with capability negotiation. That is, if a node is verified with a capability of more functions, then more modules can be loaded on demand, otherwise, only a basic module is loaded on the node.

This document defines system information, data sets and running states following the structure and definitions in IEEE 1588v2, and in [[PTP-MIB](#)].

2. Conventions used in this document

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

3. Terminology

Terminologies used in this document are extracted from [[IEEE1588](#)] and [[PTP-MIB](#)].

ARB	Arbitrary Timescale
BC	Boundary Clock
DS	Data Set
E2E	End-to-End
EUI	Extended Unique Identifier.

GPS	Global Positioning System
IANA	Internet Assigned Numbers Authority
IP	Internet Protocol
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
OC	Ordinary Clock
P2P	Peer-to-Peer
PTP	Precision Time Protocol
TAI	International Atomic Time
TC	Transparent Clock
UDP	User Datagram Protocol
UTC	Coordinated Universal Time

[4. IEEE 1588V2 YANG Model](#)

[4.1. IEEE 1588V2 YANG model hierarchy](#)

This section describes the hierarchy of IEEE 1588v2 YANG module. Query and retrieval of device wide or port specific configuration information and clock data set is described for this version, while configuration module will be described in a next version.

Query and retrieval of clock information include:

- Device level configuration information, including: clock identity, domain number, clock-running states and statistics.
- Clock data set attributes in a clock node, including: clock-current-DS, clock-parent-DS, clock-default-DS, clock-time-properties-DS, and clock-trans-default-DS and etc.
- Port specific data set attributes, including: clock-port-DS, clock-port-running, clock-port-trans-DS and clock-port-associate and etc.

```
module: ietf-yang-ptp-query
  +--rw device-ptp-query-table
  |   +--rw ptpDevicequery* [clock-identity domain-number]
  |   |   +--rw clock-identity          binary
  |   |   +--rw domain-number          uint32
  |   |   +--rw clock-current-DS
  |   |   |   +--rw steps-removed?      uint16
  |   |   |   +--rw offset-from-master? binary
  |   |   |   +--rw mean-path-delay?    binary
  |   |   +--rw clock-parent-DS
  |   |   |   +--rw parent-port-identity? binary
  |   |   |   +--rw parent-stats?        boolean
  |   |   |   +--rw offset-scaled-log-variance? int32
  |   |   |   +--rw clock-phase-change-rate? int32
  |   |   |   +--rw GM-clock-identity?    binary
  |   |   |   +--rw GM-clock-priority1?   uint8
  |   |   |   +--rw GM-clock-priority2?   uint8
  |   |   |   +--rw GM-clock-quality-class? uint32
  |   |   |   +--rw GM-clock-quality-accuracy? uint32
  |   |   |   +--rw GM-clock-quality-offset? uint16
  |   |   +--rw clock-default-DS
  |   |   |   +--rw two-step-flag?        boolean
  |   |   |   +--rw clock-identity?       binary
  |   |   |   +--rw priority1?            uint8
  |   |   |   +--rw priority2?            uint8
  |   |   |   +--rw slave-only?           boolean
  |   |   |   +--rw quality-class?        uint32
  |   |   |   +--rw quality-accuracy?     uint32
  |   |   |   +--rw quality-offset?       uint16
  |   |   +--rw clock-running
  |   |   |   +--rw device-clock-type?     enumeration
  |   |   |   +--rw clock-running-state?   enumeration
  |   |   |   +--rw clock-running-packets-sent? uint64
  |   |   |   +--rw clock-running-packets-received? uint64
  |   |   +--rw clock-time-properties-DS
  |   |   |   +--rw current-UTC-offset-valid? boolean
  |   |   |   +--rw current-UTC-offset?    uint16
  |   |   |   +--rw leap59?                boolean
  |   |   |   +--rw leap61?                boolean
  |   |   |   +--rw time-traceable?        boolean
  |   |   |   +--rw freq-traceable?        boolean
  |   |   |   +--rw PTP-timescale?         boolean
  |   |   |   +--rw source?                enumeration
  |   |   +--rw clock-trans-default-DS
  |   |   |   +--rw clock-identity?       binary
  |   |   |   +--rw num-of-ports?         uint32
  |   |   |   +--rw delay?                enumeration
```



```

|         +--rw primary-domain?  uint32
+--rw port-ptp-query-table
  +--rw ptp-port-query* [clock-identity port-number domain-
number]
    +--rw clock-identity          binary
    +--rw port-number             uint32
    +--rw domain-number           uint32
    +--rw clock-port
      | +--rw clock-port-name?    string
      | +--rw port-twostep-flag?  boolean
      | +--rw clock-port-current-peer-address?  string
      | +--rw clock-port-num-of-associated-ports?  uint32
    +--rw clock-port-DS
      | +--rw port-identity
      | | +--rw clock-identity?  binary
      | | +--rw port-number?    uint32
      | +--rw log-announcement-interval?  int64
      | +--rw announce-rct-timeout?      uint8
      | +--rw log-sync-interval?         int8
      | +--rw min-delay-req-interval?     uint8
      | +--rw peer-delay-req-interval?    uint32
      | +--rw delay-mech?                enumeration
      | +--rw peer-mean-path-delay?      binary
      | +--rw grant-duration?            uint32
      | +--rw PTP-version?               uint32
    +--rw clock-port-running
      | +--rw port-type?              enumeration
      | +--rw running-state?          enumeration
      | +--rw running-role?           enumeration
      | +--rw running-interface-index? uint32
      | +--rw running-Tx-mode?        enumeration
      | +--rw running-Rx-mode?        enumeration
      | +--rw running-packets-received? binary
      | +--rw running-packets-sent?   uint64
    +--rw clock-port-trans-DS
      | +--rw log-min-pdelay-req-int?  int32
      | +--rw faulty-flag?             boolean
      | +--rw peer-mean-path-delay?    binary
    +--rw clock-port-associate
      +--rw port-index?               uint32
      +--rw associate-address?        binary
      +--rw associate-packets-sent?   binary
      +--rw associate-packets-received? binary
      +--rw associate-in-errors?      binary
      +--rw associate-out-errors?     binary

```


4.2. IEEE 1588v2 rpc definitions

The rpc model facilitates issuing commands to a NETCONF server and obtaining a response. Rpc model defined in this document provides commands to retrieve their configuration information, data sets and running states. Configuration of 1588v2 clocks and PTP ports will be provided in a next version.

rpcs:

```

+---x read-device-ptp-query-table
| +---w input
| | +---w clock-identity?    binary
| | +---w domain-number?    uint32
| +---ro output
|   +---ro clock-current-DS
|     | +---ro steps-removed?      uint16
|     | +---ro offset-from-master?  binary
|     | +---ro mean-path-delay?    binary
|   +---ro clock-parent-DS
|     | +---ro parent-port-identity?  binary
|     | +---ro parent-stats?          boolean
|     | +---ro offset-scaled-log-variance?  int32
|     | +---ro clock-phase-change-rate?  int32
|     | +---ro GM-clock-identity?      binary
|     | +---ro GM-clock-priority1?     uint8
|     | +---ro GM-clock-priority2?     uint8
|     | +---ro GM-clock-quality-class?  uint32
|     | +---ro GM-clock-quality-accuracy?  uint32
|     | +---ro GM-clock-quality-offset?  uint16
|   +---ro clock-default-DS
|     | +---ro two-step-flag?      boolean
|     | +---ro clock-identity?     binary
|     | +---ro priority1?          uint8
|     | +---ro priority2?          uint8
|     | +---ro slave-only?         boolean
|     | +---ro quality-class?      uint32
|     | +---ro quality-accuracy?   uint32
|     | +---ro quality-offset?     uint16
|   +---ro clock-running
|     | +---ro device-clock-type?      enumeration
|     | +---ro clock-running-state?    enumeration
|     | +---ro clock-running-packets-sent?  uint64
|     | +---ro clock-running-packets-received?  uint64
|   +---ro clock-time-properties-DS
|     | +---ro current-UTC-offset-valid?  boolean
|     | +---ro current-UTC-offset?       uint16
|     | +---ro leap59?                  boolean

```



```
|      | +--ro leap61?                boolean
|      | +--ro time-traceable?        boolean
|      | +--ro freq-traceable?        boolean
|      | +--ro PTP-timescale?         boolean
|      | +--ro source?                enumeration
|      +--ro clock-trans-default-DS
|          +--ro clock-identity?      binary
|          +--ro num-of-ports?        uint32
|          +--ro delay?               enumeration
|          +--ro primary-domain?      uint32
+---x read-clock-current-DS
|   +---w input
|   |   +---w clock-identity?        binary
|   |   +---w domain-number?        uint32
|   +--ro output
|       +--ro steps-removed?          uint16
|       +--ro offset-from-master?    binary
|       +--ro mean-path-delay?       binary
+---x read-clock-parent-DS
|   +---w input
|   |   +---w clock-identity?        binary
|   |   +---w domain-number?        uint32
|   +--ro output
|       +--ro parent-port-identity?   binary
|       +--ro parent-stats?           boolean
|       +--ro offset-scaled-log-variance? int32
|       +--ro clock-phase-change-rate? int32
|       +--ro GM-clock-identity?      binary
|       +--ro GM-clock-priority1?     uint8
|       +--ro GM-clock-priority2?     uint8
|       +--ro GM-clock-quality-class?  uint32
|       +--ro GM-clock-quality-accuracy? uint32
|       +--ro GM-clock-quality-offset? uint16
+---x read-clock-default-DS
|   +---w input
|   |   +---w clock-identity?        binary
|   |   +---w domain-number?        uint32
|   +--ro output
|       +--ro two-step-flag?          boolean
|       +--ro clock-identity?        binary
|       +--ro priority1?              uint8
|       +--ro priority2?              uint8
|       +--ro slave-only?             boolean
|       +--ro quality-class?          uint32
|       +--ro quality-accuracy?       uint32
|       +--ro quality-offset?         uint16
+---x read-clock-running
```



```
| +---w input
| | +---w clock-identity?  binary
| | +---w domain-number?  uint32
| +--ro output
|   +--ro device-clock-type?          enumeration
|   +--ro clock-running-state?        enumeration
|   +--ro clock-running-packets-sent?  uint64
|   +--ro clock-running-packets-received?  uint64
+---x read-clock-time-properties-DS
| +---w input
| | +---w clock-identity?  binary
| | +---w domain-number?  uint32
| +--ro output
|   +--ro current-UTC-offset-valid?  boolean
|   +--ro current-UTC-offset?        uint16
|   +--ro leap59?                    boolean
|   +--ro leap61?                    boolean
|   +--ro time-traceable?            boolean
|   +--ro freq-traceable?            boolean
|   +--ro PTP-timescale?             boolean
|   +--ro source?                    enumeration
+---x read-clock-trans-default-DS
| +---w input
| | +---w clock-identity?  binary
| | +---w domain-number?  uint32
| +--ro output
|   +--ro clock-identity?  binary
|   +--ro num-of-ports?    uint32
|   +--ro delay?           enumeration
|   +--ro primary-domain?  uint32
+---x read-port-ptp-query-table
| +---w input
| | +---w clock-identity?  binary
| | +---w port-number?    uint32
| | +---w domain-number?  uint32
| +--ro output
|   +--ro clock-port
|   | +--ro clock-port-name?          string
|   | +--ro port-twostep-flag?        boolean
|   | +--ro clock-port-current-peer-address?  string
|   | +--ro clock-port-num-of-associated-ports?  uint32
|   +--ro clock-port-DS
|   | +--ro port-identity
|   | | +--ro clock-identity?  binary
|   | | +--ro port-number?    uint32
|   | +--ro log-announcement-interval?  int64
|   | +--ro announce-rct-timeout?      uint8
```



```

|   | +--ro log-sync-interval?          int8
|   | +--ro min-delay-req-interval?      uint8
|   | +--ro peer-delay-req-interval?     uint32
|   | +--ro delay-mech?                  enumeration
|   | +--ro peer-mean-path-delay?        binary
|   | +--ro grant-duration?              uint32
|   | +--ro PTP-version?                 uint32
| +--ro clock-port-running
|   | +--ro port-type?                   enumeration
|   | +--ro running-state?               enumeration
|   | +--ro running-role?                enumeration
|   | +--ro running-interface-index?     uint32
|   | +--ro running-Tx-mode?             enumeration
|   | +--ro running-Rx-mode?             enumeration
|   | +--ro running-packets-received?    binary
|   | +--ro running-packets-sent?        uint64
| +--ro clock-port-trans-DS
|   | +--ro log-min-pdelay-req-int?      int32
|   | +--ro faulty-flag?                 boolean
|   | +--ro peer-mean-path-delay?        binary
| +--ro clock-port-associate
|   | +--ro port-index?                  uint32
|   | +--ro associate-address?           binary
|   | +--ro associate-packets-sent?      binary
|   | +--ro associate-packets-received?  binary
|   | +--ro associate-in-errors?         binary
|   | +--ro associate-out-errors?        binary
+---x read-clock-port
| +---w input
|   | +---w clock-identity?              binary
|   | +---w port-number?                 uint32
|   | +---w domain-number?               uint32
| +--ro output
|   | +--ro clock-port-name?              string
|   | +--ro port-twostep-flag?            boolean
|   | +--ro clock-port-current-peer-address? string
|   | +--ro clock-port-num-of-associated-ports? uint32
+---x read-clock-port-DS
| +---w input
|   | +---w clock-identity?              binary
|   | +---w port-number?                 uint32
|   | +---w domain-number?               uint32
| +--ro output
|   | +--ro port-identity
|   |   | +--ro clock-identity?          binary
|   |   | +--ro port-number?             uint32
|   | +--ro log-announcement-interval?   int64

```



```

|     +---ro announce-rct-timeout?      uint8
|     +---ro log-sync-interval?         int8
|     +---ro min-delay-req-interval?    uint8
|     +---ro peer-delay-req-interval?   uint32
|     +---ro delay-mech?                enumeration
|     +---ro peer-mean-path-delay?      binary
|     +---ro grant-duration?            uint32
|     +---ro PTP-version?               uint32
+---x read-clock-port-running
|   +---w input
|   |   +---w clock-identity?    binary
|   |   +---w port-number?      uint32
|   |   +---w domain-number?    uint32
|   +---ro output
|   |   +---ro port-type?        enumeration
|   |   +---ro running-state?    enumeration
|   |   +---ro running-role?     enumeration
|   |   +---ro running-interface-index? uint32
|   |   +---ro running-Tx-mode?  enumeration
|   |   +---ro running-Rx-mode?  enumeration
|   |   +---ro running-packets-received? binary
|   |   +---ro running-packets-sent? uint64
+---x read-clock-port-trans-DS
|   +---w input
|   |   +---w clock-identity?    binary
|   |   +---w port-number?      uint32
|   |   +---w domain-number?    uint32
|   +---ro output
|   |   +---ro log-min-pdelay-req-int? int32
|   |   +---ro faulty-flag?         boolean
|   |   +---ro peer-mean-path-delay? binary
+---x read-clock-port-associate
|   +---w input
|   |   +---w clock-identity?    binary
|   |   +---w port-number?      uint32
|   |   +---w domain-number?    uint32
|   +---ro output
|   |   +---ro port-index?        uint32
|   |   +---ro associate-address? binary
|   |   +---ro associate-packets-sent? binary
|   |   +---ro associate-packets-received? binary
|   |   +---ro associate-in-errors? binary
|   |   +---ro associate-out-errors? binary

```


5. IEEE 1588v2 YANG Module

```
module ietf-yang-ptp-query{
  namespace "urn:ietf:params:xml:ns:yang:1588v2";
  prefix "ptp-query";
  organization "IETF TICTOC WG";
  contact "jiangyuanlong@huawei.com";
  description
    "This YANG module defines a data model for the configuration
    of IEEE 1588v2 devices and clocks, and also retrieval of the
    state data of IEEE 1588v2 clocks.";
  revision "2015-07-06"{
    description "Initial revision.";
    reference "draft-jxl-tictoc-1588v2-yang";
  }

  grouping device-identify-infor{
    description
      "This group bundles together all information used to
      identify a device.";
    leaf clock-identity{
      description "identify a specific PTP node.";
      config true;
      type binary {
        length "1..255";
      }
    }

    leaf domain-number{
      description
        "This object specifies the domain number used to create
        logical group of PTP devices.";
      type uint32{
        range "0..255";
      }
    }
  }

  grouping clock-current-DS-entry{
    description
      "This group bundles together all information about the
      PTP clock Current Datasets for a single device.";

    leaf steps-removed{
      description
```



```
        "specifies the distance measured by the number of
        Boundary clocks between the local clock and the
        Foreign master as indicated in the stepsRemoved
        field of Announce messages.";
    type uint16;
}
leaf offset-from-master{
    description
        "specifies the current clock dataset ClockOffset value.
        The value of the computation of the offset in time
        between a slave and a master clock.";
    type binary {
        length "1..255";
    }
}
leaf mean-path-delay{
    description
        "The mean path delay between a pair of ports as measure
        by the delay request-response mechanism.";
    type binary {
        length "1..255";
    }
}
}

grouping clock-parent-DS-entry{
    description
        "This group bundles together all information about the PTP
        clock Parent Datasets for a single device.";

    leaf parent-port-identity{
        description
            "specifies the value of portIdentity of the port on the
            master that issues the Sync messages used in
            synchronizing this clock.";
        type binary{
            length "1..256";
        }
    }
    leaf parent-stats{
        description
            "specifies the Parent Dataset ParentStats value. This
            value indicates whether the values of ParentDSOffset and
            ParentDSClockPhChRate have been measured and are valid.
            A TRUE value shall indicate valid data";
        type boolean;
    }
}
```



```
leaf offset-scaled-log-variance{
  description
    "specifies the Parent Dataset
    ParentOffsetScaledLogVariance value. This value is the
    variance of the parent clocks phase as measured by the
    local clock.";
  type int32{
    range "-128..127";
  }
}
leaf clock-phase-change-rate{
  description
    "specifies the clock's parent dataset
    ParentClockPhaseChangeRate value. This value is an
    estimate of the parent clocks phase change rate as
    measured by the slave clock.";
  type int32;
}
leaf GM-clock-identity{
  description
    "This object specifies the parent dataset Grandmaster
    clock identity";
  type binary{
    length "1..255";
  }
}
leaf GM-clock-priority1{
  description
    "This object specifies the parent dataset Grandmaster
    clock priority1.";
  type uint8;
}
leaf GM-clock-priority2{
  description
    "This object specifies the parent dataset grandmaster
    clock priority2.";
  type uint8;
}
leaf GM-clock-quality-class{
  description
    "This object specifies the parent dataset grandmaster
    clock quality class.";
  type uint32{
    range "0..255";
  }
}
leaf GM-clock-quality-accuracy{
```



```
        description
            "This object specifies the parent dataset grandmaster
            clock quality accuracy";
        type uint32;
    }
    leaf GM-clock-quality-offset{
        description
            "This object specifies the parent dataset grandmaster
            clock quality offset.";
        type uint16;
    }
}

grouping clock-default-DS-entry{
    description
        "This group bundles together all information about the
        PTP clock Default Datasets for a single device.";

    leaf two-step-flag{
        description
            "This object specifies whether the Two Step process is
            used.";
        type boolean;
    }
    leaf clock-identity{
        description
            "This object specifies the default Datasets clock
            identity";
        type binary{
            length "1..255";
        }
    }
    leaf priority1{
        description
            "This object specifies the default Datasets clock
            Priority1";
        type uint8;
    }
    leaf priority2{
        description
            "This object specifies the default Datasets clock
            Priority2";
        type uint8;
    }
    leaf slave-only{
        description "Whether the SlaveOnly flag is set";
        type boolean;
    }
}
```



```
}
leaf quality-class{
  description
    "This object specifies the default dataset Quality
    Class.";
  type uint32{
    range "0..255";
  }
}
leaf quality-accuracy{
  description
    "This object specifies the default dataset Quality
    Accuracy.";
  type uint32;
}
leaf quality-offset{
  description
    "This object specifies the default dataset Quality
    offset.";
  type uint16;
}
}

grouping clock-running-entry{
  description
    "This group bundles together all information about the
    PTP clock Running Datasets for a single device.";

  leaf device-clock-type{
    description "This object specifies the clock type.";
    type enumeration{
      enum oc{
        value 0;
        description "ordinary clock ";
      }
      enum bc{
        value 1;
        description "boundary clock";
      }
      enum e2etc{
        value 2;
        description "End-to-end transparent clock";
      }
      enum p2ptc{
        value 3;
        description "Peer-to-peer transparent clock ";
      }
    }
  }
}
```



```
    enum e2etcoc{
        value 4;
        description
            "End-to-End transparent clock ordinary clock";
    }
    enum p2ptcoc{
        value 5;
        description
            "peer-to-peer transparent clock ordinary clock ";
    }
    enum tcandbc{
        value 6;
        description "transparent clock and boundary clock";
    }
}
}

leaf clock-running-state{
    description
        "specifies the Clock state returned by PTP engine which
        was described earlier.";
    type enumeration{
        enum freerun{
            value 1;
            description
                "Applies to a slave device that is not locked to a
                master. This is the initial state a slave starts
                out with when it
                is not getting any PTP packets from the master or
                because of some other input error (erroneous
                packets, etc).";
        }
        enum holdover{
            value 2;
            description
                "In this state the slave device is locked to a
                master but communication with the master has been
                lost or the timestamps in the ptp packets are
                incorrect. Since the slave was locked to the
                master, it can run in this state, with similar
                accuracy for some time. If communication with the
                master is not restored for an extended period
                (dependent on the clock implementation), the
                device should move to the FREERUN state.";
        }
        enum acquiring{
            value 3;
```



```
        description
            "The slave device is receiving packets from a
            master and is trying to acquire a lock.";
    }
    enum frequencyLocked{
        value 4;
        description
            "Slave device is locked to the Master with respect
            to frequency, but not phase aligned";
    }
    enum phaseAligned{
        value 5;
        description
            "Locked to the master with respect to frequency
            and phase.";
    }
}
}
leaf clock-running-packets-sent{
    description
        "This object specifies the total number of all packet
        Unicast and multicast that have been sent out for this
        clock in this domain for this type.";
    type uint64;
}
leaf clock-running-packets-received{
    description
        "This object specifies the total number of all packet
        Unicast and multicast that have been received for this
        clock in this domain for this type.";
    type uint64;
}
}

grouping clock-time-properties-DS-entry{
    description
        "This group bundles together all information about the
        PTP clock time properties datasets for a single device.";

    leaf current-UTC-offset-valid{
        description
            "This object specifies the timeproperties dataset value
            of whether current UTC offset is valid.";
        type boolean;
    }
    leaf current-UTC-offset{
        description
```



```
        "This object specifies the timeproperties dataset value
        of current UTC offset. In PTP systems whose epoch is the
        PTP epoch, the value of
        timePropertiesDS.currentUtcOffset is the offset between
        TAI and UTC; otherwise the value has no meaning. The
        value shall be in units of seconds. The initialization
        value shall be selected as follows: a) If the
        timePropertiesDS.ptpTimescale is TRUE, the value is the
        value obtained from a primary reference if the value is
        known at the time of initialization, else, b) The value
        shall be the current number of leap seconds (7.2.3)
        when the node is designed.";
    type uint16;
}
leaf leap59{
    description
        "This object specifies the Leap59 value in the clock
        Current Dataset.";
    type boolean;
}
leaf leap61{
    description
        "This object specifies the Leap61 value in the clock
        Current Dataset.";
    type boolean;
}
leaf time-traceable{
    description
        "This object specifies the Timetraceable value in the
        clock Current Dataset.";
    type boolean;
}
leaf freq-traceable{
    description
        "This object specifies the Frequency Traceable value in
        the clock Current Dataset.";
    type boolean;
}
leaf PTP-timescale{
    description
        "This object specifies the PTP Timescale value in the
        clock Current Dataset.";
    type boolean;
}
leaf source{
    description
```



```
        "This object specifies the Timesource value in the clock
        Current Dataset.";
    type enumeration{
        enum atomicClock{
            value 16;
        }
        enum gps{
            value 32;
        }
        enum terrestrialRadio{
            value 48;
        }
        enum ptp{
            value 64;
        }
        enum ntp{
            value 80;
        }
        enum handSet{
            value 96;
        }
        enum other{
            value 144;
        }
        enum internalOscillator{
            value 160;
        }
    }
}
}

grouping clock-trans-default-DS-entry{
    description
        "This group bundles together all information about the
        PTP Transparent clock Default Datasets for a single
        device.";

    leaf clock-identity{
        description
            "This object specifies the value of the clockIdentity
            attribute of the local clock.";
        type binary{
            length "0..255";
        }
    }
    leaf num-of-ports{
        description
```



```
        "This object specifies the number of PTP ports of the
        device.";
    type uint32;
}
leaf delay{
    description
        "This object, if the transparent clock is an end-to-end
        transparent clock, has the value shall be E2E; If the
        transparent clock is a peer-to-peer transparent clock,
        the value shall be P2P.";
    type enumeration{
        enum E2E{
            value 1;
            description
                "The port is configured to use the delay request-
                response mechanism.";
        }
        enum P2P{
            value 2;
            description
                "The port is configured to use the peer delay
                mechanism.";
        }
        enum DISABLED{
            value 254;
            description
                "The port does not implement the delay mechanism.";
        }
    }
}
leaf primary-domain{
    description
        "This object specifies the value of the primary
        syntonization domain. The initialization value shall be
        0.";
    type uint32{
        range "0..255";
    }
}
}

container device-ptp-query-table{
    list ptpDevicequery{
        description "";

        key "clock-identity domain-number";
        uses device-identify-infor;
```



```
    container clock-current-DS{
        uses clock-current-DS-entry;
    }
    container clock-parent-DS{
        uses clock-parent-DS-entry;
    }
    container clock-default-DS{
        uses clock-default-DS-entry;
    }
    container clock-running{
        uses clock-running-entry;
    }
    container clock-time-properties-DS{
        uses clock-time-properties-DS-entry;
    }
    container clock-trans-default-DS{
        uses clock-trans-default-DS-entry;
    }
}
}

rpc read-device-ptp-query-table{
    description
        "Read all device-level PTP informations of a specific
        device.";
    input{
        uses device-identify-infor;
    }
    output{
        container clock-current-DS{
            uses clock-current-DS-entry;
        }
        container clock-parent-DS{
            uses clock-parent-DS-entry;
        }
        container clock-default-DS{
            uses clock-default-DS-entry;
        }
        container clock-running{
            uses clock-running-entry;
        }
        container clock-time-properties-DS{
            uses clock-time-properties-DS-entry;
        }
        container clock-trans-default-DS{
            uses clock-trans-default-DS-entry;
        }
    }
}
```



```
    }  
  }  
  rpc read-clock-current-DS {  
    description  
      "Reads the information about the PTP clock Current  
      Datasets for all domains.";  
    input {  
      uses device-identify-infor;  
    }  
    output {  
      uses clock-current-DS-entry;  
    }  
  }  
  
  rpc read-clock-parent-DS {  
    description  
      "Reads the information about the PTP clock Parent  
      Datasets for all domains.";  
    input {  
      uses device-identify-infor;  
    }  
    output {  
      uses clock-parent-DS-entry;  
    }  
  }  
  
  rpc read-clock-default-DS {  
    description  
      "Reads the information about the PTP clock Parent  
      Datasets for all domains.";  
    input {  
      uses device-identify-infor;  
    }  
    output {  
      uses clock-default-DS-entry;  
    }  
  }  
  
  rpc read-clock-running{  
    description  
      "Reads the clock Running PTP information for a device.";  
    input {  
      uses device-identify-infor;  
    }  
    output {  
      uses clock-running-entry;  
    }  
  }
```



```
    }

    rpc read-clock-time-properties-DS{
      description
        "Reads the information about the PTP clock
        TimeProperties Datasets for all domains.";
      input {
        uses device-identify-infor;
      }
      output {
        uses clock-time-properties-DS-entry;
      }
    }

    rpc read-clock-trans-default-DS {
      description
        "Reads the information about the PTP clock Parent
        Datasets for all domains.";
      input {
        uses device-identify-infor;
      }
      output {
        uses clock-trans-default-DS-entry;
      }
    }

    grouping port-identify-info{
      description "";

      leaf clock-identity {
        description "identify a specific PTP node.";
        config true;
        type binary {
          length "1..255";
        }
      }
    }

    leaf port-number{
      description
        "This object specifies the PTP Portnumber for this
        port.";
      type uint32{
        range "0..65535";
      }
    }
  }

  leaf domain-number{
```



```
        description
            "This object specifies the domain number used to create
            logical group of PTP devices.";
        type uint32{
            range "0..255";
        }
    }
}

grouping clock-port-entry{
    description
        "This group bundles together basic information about a
        single clock port.";

    leaf clock-port-name{
        description
            "This object specifies the PTP clock port name
            configured on the router.";
        type string;
    }

    leaf port-twostep-flag{
        description
            "This object specifies that two-step clock operation
            between the PTP master and slave device is enabled.";
        type boolean;
    }

    leaf clock-port-current-peer-address{
        description
            "This object specifies the current peer's network
            address used for PTP communication.";
        type string;
    }

    leaf clock-port-num-of-associated-ports{
        description
            "This object specifies - For a master port - the number
            of PTP slave sessions (peers) associated with this PTP
            port. For a slave port - the number of masters available
            to this slave port (might or might not be peered).";
        type uint32;
    }
}

grouping clock-port-DS-entry{
    description
```



```
        "This group bundles together all information about the
        clock ports dataset for a single clock port.";

    container port-identity{
        description
            "This object specifies the PTP clock port Identity,
            composed of clock-identity and portNumber.";
        leaf clock-identity {
            description "identify a specific PTP node.";
            config true;
            type binary {
                length "1..255";
            }
        }

        leaf port-number{
            description
                "This object specifies the PTP Portnumber for this
                port.";
            type uint32{
                range "0..65535";
            }
        }
    }

    leaf log-announcement-interval{
        description
            "This object specifies the Announce message transmission
            interval associated with this clock port.";
        type int64;
    }

    leaf announce-rct-timeout{
        description
            "This object specifies the Announce receipt timeout
            associated with this clock port.";
        type uint8;
    }

    leaf log-sync-interval{
        description
            "This object specifies the Sync message transmission
            interval.";
        type int8;
    }

    leaf min-delay-req-interval{
```



```
    description
        "This object specifies the Delay_Req message
        transmission interval.";
    type uint8;
}

leaf peer-delay-req-interval{
    description
        "This object specifies the Pdelay_Req message
        transmission interval.";
    type uint32;
}

leaf delay-mech{
    description
        "specifies the delay mechanism used. If the clock is an
        end-to-end clock, the value of the is e2e, else if the
        clock is a peer to-peer clock, the value shall be p2p.";
    type enumeration{
        enum E2E{
            value 01;
            description
                "The port is configured to use the delay request-
                response mechanism.";
        }
        enum P2P{
            value 02;
            description
                "The port is configured to use the peer delay
                mechanism.";
        }
        enum DISABLED{
            value 254;
            description
                "The port does not implement the delay mechanism.";
        }
    }
}

leaf peer-mean-path-delay{
    description
        "This object specifies the peer meanPathDelay.";
    type binary{
        length "1..256";
    }
}
```



```
    leaf grant-duration{
      description
        "This object specifies the grant duration allocated by
        the master.";
      type uint32;
    }

    leaf PTP-version{
      description
        "This object specifies the PTP version being used.";
      type uint32;
    }
  }

  grouping clock-port-running-entry{
    description
      "This group bundles together all running dataset
      information about a single clock port.";

    leaf port-type{
      description
        "when the clock is configured in TCandBC mode, its port
        can be configured to TC or BC.";
      config true;
      when "device-clock-type='tcandbc' ";
      type enumeration{
        enum boundaryClock {
          value 2;
          description "configured as BC";
        }
        enum transparentClock {
          value 3;
          description "configured as TC";
        }
      }
    }
  }

  leaf running-state{
    description
      "This object specifies the port state returned by PTP
      engine. ";
    type enumeration{
      enum initializing{
        value 1;
        description
          "In this state a port initializes its data sets,
          hardware, and communication facilities.";
      }
    }
  }
}
```



```
}
enum faulty{
    value 2;
    description "The fault state of the protocol.";
}
enum disabled{
    value 3;
    description
        "The port shall not place any messages on its
        communication path.";
}
enum listening{
    value 4;
    description
        "The port is waiting for the
        announceReceiptTimeout to expire or to receive an
        Announce message from a master.";
}
enum preMaster{
    value 5;
    description
        "The port shall behave in all respects as though
        it were in the MASTER state except that it shall
        not place any messages on its communication path
        except for Pdelay_Req, Pdelay_Resp,
        Pdelay_Resp_Follow_Up, signaling, or management
        messages.";
}
enum master{
    value 6;
    description "The port is behaving as a master port.";
}
enum passive{
    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 uncalibrated{
    value 8;
    description
        "The local port is preparing to synchronize to the
        master port.";
}
```



```
        enum slave{
            value 9;
            description
                "The port is synchronizing to the selected master
                port.";
        }
    }
}
leaf running-role{
    description "This object specifies the Clock Role.";
    type enumeration{
        enum master{
            value 1;
            description
                "A clock that is the source of time to which all
                other clocks on that path synchronize.";
        }
        enum slave{
            value 2;
            description
                "A clock which synchronizes to another clock
                (master).";
        }
    }
}
leaf running-interface-index{
    description
        "This object specifies the interface on the router being
        used by the PTP Clock for PTP communication.";
    type uint32;
}

leaf running-Tx-mode{
    description "specifies the clock transmission mode ";
    type enumeration{
        enum unicast{
            value 1;
            description "Using unicast communication channel.";
        }
        enum multicast{
            value 2;
            description "Using Multicast communication channel.";
        }
        enum multicastmix{
            value 3;
            description
                "Using multicast-unicast communication channel.";
        }
    }
}
```



```
    }
  }
}

leaf running-Rx-mode{
  description "This object specifie the clock receive mode ";
  type enumeration{
    enum unicast{
      value 1;
      description "Using unicast communication channel.";
    }
    enum multicast{
      value 2;
      description "Using Multicast communication channel.";
    }
    enum multicastmix{
      value 3;
      description
        "Using multicast-unicast communication channel.";
    }
  }
}

leaf running-packets-received{
  description
    "This object specifies the packets received on the clock
    port (cummulative).";
  type binary{
    length "1..255";
  }
}

leaf running-packets-sent{
  description
    "This object specifies the packets sent on the clock
    port (cummulative).";
  type uint64;
}

grouping clock-port-trans-DS-entry{
  description
    "This group bundles together all clock port Transparent
    dataset information about a single clock port.";

  leaf log-min-pdelay-req-int{
    description
      "This object specifies the value of the logarithm to the
      base 2 of the minPdelayReqInterval.";
```



```
        type int32{
            range "-128..127";
        }
    }
    leaf faulty-flag{
        description
            "This object specifies the value TRUE if the port is
            faulty and FALSE if the port is operating normally.";
        type boolean;
    }
    leaf peer-mean-path-delay{
        description
            "This object specifies, (if the delayMechanism used is
            P2P) the value is the estimate of the current one-way
            propagation delay, i.e., <meanPathDelay> on the link
            attached to this port, computed using the peer delay
            mechanism. If the value of the delayMechanism used is
            E2E, then the value will be zero.";
        type binary{
            length "1..255";
        }
    }
}

grouping clock-port-associate-entry{
    description
        "This group bundles together all information about a
        single associated port for the given clockport.";

    leaf port-index{
        description
            "This object specifies the associated port's serial
            number in the current port's context.";
        type uint32{
            range "1..65535";
        }
    }
}

leaf associate-address{
    description
        "This object specifies the peer port's network address
        used for PTP communication.";
    type binary{
        length "1..255";
    }
}

leaf associate-packets-sent{
```



```
        description
            "The number of packets sent to this peer port from the
            current port.";
        type binary{
            length "1..255";
        }
    }
    leaf associate-packets-received{
        description
            "The number of packets received from this peer port by
            the current port.";
        type binary{
            length "1..255";
        }
    }
    leaf associate-in-errors{
        description
            "This object specifies the input errors associated with
            the peer port.";
        type binary{
            length "1..255";
        }
    }
    leaf associate-out-errors{
        description
            "This object specifies the output errors associated with
            the peer port.";
        type binary{
            length "1..255";
        }
    }
}

container port-ptp-query-table{
    list ptp-port-query{
        description "";

        key "clock-identity port-number domain-number";
        uses port-identify-info;
        container clock-port{
            uses clock-port-entry;
        }
        container clock-port-DS{
            uses clock-port-DS-entry;
        }
        container clock-port-running{
            uses clock-port-running-entry;
        }
    }
}
```



```
    }
    container clock-port-trans-DS{
      uses clock-port-trans-DS-entry;
    }
    container clock-port-associate{
      uses clock-port-associate-entry;
    }
  }
}

rpc read-port-ptp-query-table{
  description
    "Read all port-level ptp information of a specific port
    on a specific device in a specific domain.";
  input{
    uses port-identify-info;
  }
  output{
    container clock-port{
      uses clock-port-entry;
    }
    container clock-port-DS{
      uses clock-port-DS-entry;
    }
    container clock-port-running{
      uses clock-port-running-entry;
    }
    container clock-port-trans-DS{
      uses clock-port-trans-DS-entry;
    }
    container clock-port-associate{
      uses clock-port-associate-entry;
    }
  }
}

rpc read-clock-port {
  description
    "Reads the information about the PTP clock Parent
    Datasets for all domains.";
  input {
    uses port-identify-info;
  }
  output {
    uses clock-port-entry;
  }
}
```



```
rpc read-clock-port-DS{
  description
    "Reads the information about the PTP clock Parent
    Datasets for all domains.";
  input {
    uses port-identify-info;
  }
  output {
    uses clock-port-DS-entry;
  }
}

rpc read-clock-port-running{
  description
    "Reads the information about the PTP clock Parent
    Datasets for all domains.";
  input {
    uses port-identify-info;
  }
  output {
    uses clock-port-running-entry;
  }
}

rpc read-clock-port-trans-DS{
  description
    "Reads the information about the PTP clock Parent
    Datasets for all domains.";
  input {
    uses port-identify-info;
  }
  output {
    uses clock-port-trans-DS-entry;
  }
}

rpc read-clock-port-associate{
  description
    "Reads the information about the PTP clock Parent
    Datasets for all domains.";
  input {
    uses port-identify-info;
  }
  output {
    uses clock-port-associate-entry;
  }
}
}
```


6. Security Considerations

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

Some data nodes defined in this YANG module are writable, and any changes to them may adversely impact a synchronization network.

7. IANA Considerations

This document registers a URI in the IETF XML registry, and the following registration is requested to be made:

URI: urn:ietf:params:xml:ns:yang:1588v2

This document registers a YANG module in the YANG Module Names:

name: 1588v2 namespace: urn:ietf:params:xml:ns:yang:1588v2

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997
- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF) ", [RFC 6020](#), October 2010
- [RFC6991] Schoenwaelder, J., "Common YANG Data Types", [RFC 6991](#), July 2013
- [IEEE1588] IEEE, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Std 1588-2008, July 2008

8.2. Informative References

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

- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", [RFC 6242](#), June 2011
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", [RFC 6536](#), March 2012
- [PTP-MIB] Shankarkumar, V., Montini, L., Frost, T., and Dowd, G., "Precision Time Protocol Version 2 (PTPv2) Management Information Base", [draft-ietf-tictoc-ntp-mib-07](#), Work in progress

9. Acknowledgments

TBD

Authors' Addresses

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

Xian Liu
Huawei Technologies Co., Ltd.
Bantian, Longgang district
Shenzhen 518129, China

Jinchun Xu
Huawei Technologies Co., Ltd.
Bantian, Longgang district
Shenzhen 518129, China