

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: August 1, 2015

N. Wu  
A. Kumar S N  
Huawei  
January 28, 2015

## **A YANG Data Model for NTP draft-wu-ntp-ntp-cfg-00**

### Abstract

This document defines a YANG data model for Network Time Protocol implementations. The data model includes configuration data and state data.

### Requirements Language

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](#)].

### 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 <http://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 August 1, 2015.

### 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.

## Table of Contents

|                      |                                     |                    |
|----------------------|-------------------------------------|--------------------|
| <a href="#">1.</a>   | Introduction . . . . .              | <a href="#">2</a>  |
| <a href="#">1.1.</a> | Terminology . . . . .               | <a href="#">2</a>  |
| <a href="#">1.2.</a> | Tree Diagrams . . . . .             | <a href="#">3</a>  |
| <a href="#">2.</a>   | NTP data model . . . . .            | <a href="#">3</a>  |
| <a href="#">3.</a>   | Relationship to NTPv4-MIB . . . . . | <a href="#">5</a>  |
| <a href="#">4.</a>   | NTP YANG Module . . . . .           | <a href="#">7</a>  |
| <a href="#">5.</a>   | IANA Considerations . . . . .       | <a href="#">21</a> |
| <a href="#">6.</a>   | Security Considerations . . . . .   | <a href="#">21</a> |
| <a href="#">7.</a>   | Acknowledgments . . . . .           | <a href="#">22</a> |
| <a href="#">8.</a>   | References . . . . .                | <a href="#">22</a> |
| <a href="#">8.1.</a> | Normative References . . . . .      | <a href="#">22</a> |
| <a href="#">8.2.</a> | Informative References . . . . .    | <a href="#">23</a> |
|                      | Authors' Addresses . . . . .        | <a href="#">23</a> |

## [1.](#) Introduction

This document defines a YANG [[RFC6020](#)] data model for Network Time Protocol [[RFC5905](#)] implementations.

The data model covers configuration of system parameters of NTP, such as access rules, authentication and VRF binding, and also associations of NTP in different modes and parameters of per-interface. It also provides information about running state of NTP implementations.

### [1.1.](#) Terminology

The following terms are defined in [[RFC6020](#)]:

- o configuration data
- o data model
- o module
- o state data

The terminology for describing YANG data models is found in [[RFC6020](#)].



## 1.2. Tree Diagrams

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

- o Brackets "[" and "]" enclose list keys.
- o Abbreviations before data node names: "rw" means configuration data (read-write), and "ro" means state data (read-only).
- o Symbols after data node names: "?" means an optional node, "!" means a presence container, and "\*" denotes a list and leaf-list.
- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

## 2. NTP data model

This document defines the YANG module "ietf-ntp", which has the following structure:

```
module: ietf-ntp
  +--rw ntp-cfg!
  |   +--rw ntp-enabled?          boolean
  |   +--rw refclock-master
  |   |   +--rw master?          boolean
  |   |   +--rw master-stratum?  ntp-stratum
  |   +--rw authentication!
  |   |   +--rw auth-enabled?     boolean
  |   |   +--rw trusted-key?      uint32
  |   |   +--rw authentication-keys* [key-id]
  |   |       +--rw key-id        uint32
  |   |       +--rw algorithm?    enumeration
  |   |       +--rw password?     union
  |   +--rw access-rules
  |   |   +--rw access-rule* [access-mode]
  |   |       +--rw access-mode    enumeration
  |   |       +--rw acl-number
  |   |           +--rw (acl-type)?
  |   |               +--:(ipv4)
  |   |                   | +--rw acl-number-ipv4?  uint16
  |   |                   +--:(ipv6)
  |   |                       +--rw acl-number-ipv6?  uint16
  |   +--rw associations
```



```

| | +--rw peers
| | | +--rw peer* [address vrf]
| | |   +--rw version?  ntp-version
| | |   +--rw address    inet:ip-address
| | |   +--rw key-id?    leafref
| | |   +--rw minpoll?   ntp-minpoll
| | |   +--rw maxpoll?   ntp-maxpoll
| | |   +--rw prefer?    boolean
| | |   +--rw burst?     boolean
| | |   +--rw iburst?    boolean
| | |   +--rw vrf        string
| | |   +--rw source?    leafref
| | +--rw servers
| | | +--rw server* [address vrf]
| | |   +--rw version?  ntp-version
| | |   +--rw address    inet:ip-address
| | |   +--rw key-id?    leafref
| | |   +--rw minpoll?   ntp-minpoll
| | |   +--rw maxpoll?   ntp-maxpoll
| | |   +--rw prefer?    boolean
| | |   +--rw burst?     boolean
| | |   +--rw iburst?    boolean
| | |   +--rw vrf        string
| | |   +--rw source?    leafref
| +--rw ntp-interfaces
| | +--rw ntp-interface* [ntp-ifname]
| | | +--rw ntp-ifname      leafref
| | | +--rw multicast-client
| | | | +--rw multicast-client-address?  union
| | | +--rw multicast-server
| | | | +--rw multicast-server-address?  inet:ip-address
| | | | +--rw multicast-server-ttl?      uint8
| | | | +--rw multicast-server-version?  ntp-version
| | | | +--rw multicast-server-keyid?    leafref
| | | +--rw broadcast-client
| | | | +--rw broadcast-client-enabled?  boolean
| | | +--rw broadcast-server
| | | | +--rw broadcast-server-version?  ntp-version
| | | | +--rw broadcast-server-keyid?    leafref
+--ro ntp-state
+--ro system-status
| +--ro clock-state?      enumeration
| +--ro clock-stratum?    ntp-stratum
| +--ro clock-refid?      union
| +--ro nominal-freq?     decimal64
| +--ro actual-freq?      decimal64
| +--ro clock-precision?  uint8
| +--ro clock-offset?     decimal64

```



```

| +--ro root-delay?          decimal64
| +--ro root-dispersion?     decimal64
| +--ro peer-dispersion?     decimal64
| +--ro reference-time?      string
| +--ro sync-state?          enumeration
+--ro associations-status
| +--ro association-status* [association-source]
|   +--ro association-source      union
|   +--ro association-stratum?    ntp-stratum
|   +--ro association-refid?      union
|   +--ro association-reach?      uint8
|   +--ro association-poll?       uint8
|   +--ro association-now?        uint32
|   +--ro association-offset?     decimal64
|   +--ro association-delay?      decimal64
|   +--ro association-dispersion? decimal64
|   +--ro association-sent?       uint32
|   +--ro association-sent-fail?  uint32
|   +--ro association-received?   uint32
|   +--ro association-dropped?    uint32
+--ro ntp-statistics
  +--ro packet-sent?            uint32
  +--ro packet-sent-fail?       uint32
  +--ro packet-received?        uint32
  +--ro packet-dropped?         uint32

```

This data model defines two primary containers, one for NTP configuration and the other is for NTP running state. The NTP configuration container includes data nodes for access rules, authentication, associations and interfaces. In the NTP running state container, there are data nodes for system status and associations.

### 3. Relationship to NTPv4-MIB

If the device implements the NTPv4-MIB [[RFC5907](#)], data nodes in container ntp-cfg and ntp-state from YANG module can be mapped to table entries in NTPv4-MIB.

The following tables list the YANG data nodes with corresponding objects in the NTPv4-MIB.



|   |                              |   |                         |   |
|---|------------------------------|---|-------------------------|---|
| + | -----                        | + | -----                   | + |
|   | YANG data nodes in /ntp-cfg/ |   | NTPv4-MIB objects       |   |
| + | -----                        | + | -----                   | + |
|   | ntp-enabled                  |   | ntpEntStatusCurrentMode |   |
| + | -----                        | + | -----                   | + |

|   |                                      |   |                     |   |
|---|--------------------------------------|---|---------------------|---|
| + | -----                                | + | -----               | + |
|   | YANG data nodes in                   |   | NTPv4-MIB objects   |   |
|   | /ntp-cfg/associations/peers/peer     |   |                     |   |
|   | /ntp-cfg/associations/servers/server |   |                     |   |
| + | -----                                | + | -----               | + |
|   | address                              |   | ntpAssocAddressType |   |
|   |                                      |   | ntpAssocAddress     |   |
| + | -----                                | + | -----               | + |

#### YANG NTP Configuration Data Nodes and Related NTPv4-MIB Objects

|   |                          |   |                                 |   |
|---|--------------------------|---|---------------------------------|---|
| + | -----                    | + | -----                           | + |
|   | YANG data nodes in       |   | NTPv4-MIB objects               |   |
|   | /ntp-state/system-status |   |                                 |   |
| + | -----                    | + | -----                           | + |
|   | clock-state              |   | ntpEntStatusCurrentMode         |   |
|   | clock-stratum            |   | ntpEntStatusStratum             |   |
|   | clock-refid              |   | ntpEntStatusActiveRefSourceId   |   |
|   |                          |   | ntpEntStatusActiveRefSourceName |   |
|   | clock-precision          |   | ntpEntTimePrecision             |   |
|   | clock-offset             |   | ntpEntStatusActiveOffset        |   |
|   | root-dispersion          |   | ntpEntStatusDispersion          |   |
| + | -----                    | + | -----                           | + |

|   |                                 |   |                           |   |
|---|---------------------------------|---|---------------------------|---|
| + | -----                           | + | -----                     | + |
|   | YANG data nodes in              |   | NTPv4-MIB objects         |   |
|   | /ntp-state/associations-status/ |   |                           |   |
|   | association-status/             |   |                           |   |
| + | -----                           | + | -----                     | + |
|   | association-source              |   | ntpAssocAddressType       |   |
|   |                                 |   | ntpAssocAddress           |   |
|   | association-stratum             |   | ntpAssocStratum           |   |
|   | association-refid               |   | ntpAssocRefId             |   |
|   | association-offset              |   | ntpAssocOffset            |   |
|   | association-delay               |   | ntpAssocStatusDelay       |   |
|   | association-dispersion          |   | ntpAssocStatusDispersion  |   |
|   | association-sent                |   | ntpAssocStatOutPkts       |   |
|   | association-received            |   | ntpAssocStatInPkts        |   |
|   | association-dropped             |   | ntpAssocStatProtocolError |   |
| + | -----                           | + | -----                     | + |

#### YANG NTP State Data Nodes and Related NTPv4-MIB Objects



#### 4. NTP YANG Module

```
//<CODE BEGINS> file "ietf-ntp@2015-01-28.yang"

module ietf-ntp {

    namespace "urn:ietf:params:xml:ns:yang:ietf-ntp";

    prefix "ntp";

    import ietf-yang-types {
        prefix "yang";
    }

    import ietf-inet-types {
        prefix "inet";
    }

    import ietf-interfaces {
        prefix "if";
    }

    import ietf-ip {
        prefix "ip";
    }

    organization
        "IETF NTP (Network Time Protocol) Working Group";

    contact
        "WG Web: <http://tools.ietf.org/wg/ntp/>
        WG List: <mailto:ntpwg@lists.ntp.org>
        WG Chair: Karen O'Donoghue
                <mailto:odonoghue@isoc.org>
        Editor:   Eric Wu
                <mailto:eric.wu@huawei.com>";

    description
        "This YANG module defines essential components for the management
        of a routing subsystem.

        Copyright (c) 2014 IETF Trust and the persons identified as
        authors of the code. All rights reserved.

        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject to
        the license terms contained in, the Simplified BSD License set
        forth in Section 4.c of the IETF Trust's Legal Provisions
```



Relating to IETF Documents

(<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

```
revision 2015-01-28 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: A YANG Data Model for NTP Management";
}

/* Typedef Definitions */
typedef ntp-stratum {
  type uint8;
  description
    "The level of each server in the hierarchy is defined by
    a stratum number. Primary servers are assigned stratum one;
    secondary servers at each lower level are assigned stratum
    numbers one greater than the preceding level";
}

typedef ntp-version {
  type uint8 {
    range "1..4";
  }
  default "3";
  description
    "The current NTP version supported by corresponding association.";
}

typedef ntp-minpoll {
  type uint8 {
    range "4..17";
  }
  default "6";
  description
    "The minimul poll interval for this NTP association.";
}

typedef ntp-maxpoll {
  type uint8 {
    range "4..17";
  }
  default "10";
```



```
    description
        "The maximul poll interval for this NTP association.";
}

typedef multicast-client-v4address {
    type inet:ipv4-address;
    default "224.0.1.1";
    description
        "The IPv4 address for NTP multicast client.";
}

typedef multicast-client-v6address {
    type inet:ipv6-address;
    default "FF0E::0101";
    description
        "The IPv6 address for NTP multicast client.";
}

/* Groupings */
grouping authentication-key {
    description
        "To define an authentication key for a NTP time source.";
    leaf key-id {
        type uint32 {
            range "1..max";
        }
        description
            "Authentication key identifier.";
    }
    leaf algorithm {
        type enumeration {
            enum md5 {
                description
                    "Message Digest 5 (MD5) algorithm.";
            }
            enum hmac-sha256 {
                description
                    "Secure Hash Algorithm 256 algorithm.";
            }
        }
        description
            "Authentication algorithm.";
    }
    leaf password {
        type union {
            type string {
                length "1..255";
            }
        }
    }
}
```



```
        type string {
            length "20..392";
        }
    }
    description
        "Clear or encrypted mode for password text.";
}

grouping association-param {
    description
        "To define parameters for a NTP associations.";
    leaf version {
        type ntp-version;
        description
            "NTP version.";
    }
    leaf address {
        type inet:ip-address;
        description
            "The IP address of this association.";
    }
    leaf key-id {
        type leafref {
            path "/ntp:ntp-cfg/ntp:authentication/ntp:authentication-keys/
ntp:key-id";
        }
        description
            "Authentication key id referenced in this association.";
    }
    leaf minpoll {
        type ntp-minpoll;
        description
            "The minimul poll interval used in this association.";
    }
    leaf maxpoll {
        type ntp-maxpoll;
        description
            "The maximul poll interval used in this association.";
    }
    leaf prefer {
        type boolean;
        default "false";
        description
            "Whether this association is preferred.";
    }
    leaf burst {
        type boolean;
```

```
default "false";
```

```
        description
            "Sends a series of packets instead of a single packet
            within each synchronization interval to achieve
            faster synchronization.";
    }
    leaf iburst {
        type boolean;
        default "false";
        description
            "Sends a series of packets instead of a single packet
            within the initial synchronization interval to achieve
            faster initial synchronization.";
    }
    leaf vrf {
        type string;
        description
            "The VRF instance this association binded to.";
    }
    leaf source {
        type leafref {
            path "/if:interfaces/if:interface/if:name";
        }
        description
            "The interface whose ip address this association used as source
address.";
    }
}
```

```
/* Configuration data nodes */
```

```
container ntp-cfg {
    presence
        "Enables NTP unless the 'ntp-enabled' leaf
        (which defaults to 'true') is set to 'false'";
    description
        "Configuration parameters for NTP.";
    leaf ntp-enabled {
        type boolean;
        default true;
        description
            "Controls whether NTP is enabled or disabled on this device.";
    }
    container refclock-master {
        leaf master {
            type boolean;
            default false;
            description
                "Use its own NTP master clock to synchronize with peers when
```

```
true.";
    }
```

```
    leaf master-stratum {
        type ntp-stratum;
        default "16";
        description
            "Use its own NTP master clock to synchronize with peers when
true.";
    }
}

container authentication {
    presence
        "Enables NTP authentication when the 'auth-enabled'
        leaf is set to 'true'.";
    description
        "Configuration of authentication.";
    leaf auth-enabled {
        type boolean;
        default false;
        description
            "Controls whether NTP authentication is enabled or disabled
            on this device.";
    }
    leaf trusted-key {
        type uint32;
        description
            "The key trusted by NTP.";
    }
    list authentication-keys {
        key "key-id";
        uses authentication-key;
    }
}

}

container access-rules {
    list access-rule {
        key "access-mode";
        leaf access-mode {
            type enumeration {
                enum peer {
                    description
                        "Sets the fully access authority.
                        Both time request and control query can be performed
                        on the local NTP service, and the local clock can be
                        synchronized to the remote server.";
                }
            }
        }
    }
}
```

}

```
enum server {
  description
    "Enables the server access and query.
    Both time requests and control query can be performed
    on the local NTP service, but the local clock cannot be
    synchronized to the remote server.";
}
enum synchronization {
  description
    "Enables the server to access.
    Only time request can be performed
    on the local NTP service.";
}
enum query {
  description
    "Sets the maximum access limitation.
    Control query can be performed only
    on the local NTP service.";
}
}
description
  "NTP access mode.";
}
container acl-number {
  choice acl-type {
    case ipv4 {
      leaf acl-number-ipv4 {
        type uint16;
        description "IPv4 acl number.";
      }
    }
    case ipv6 {
      leaf acl-number-ipv6 {
        type uint16;
        description "IPv6 acl number.";
      }
    }
  }
}
}
}
}
}

container associations {
  container peers {
    description
      "Peer associations.";
    list peer {
      key "address vrf";
```



```
        uses association-param;
    }
}
container servers {
    description
        "Sever associations.";
    list server {
        key "address vrf";
        uses association-param;
    }
}
}

container ntp-interfaces {
    description "Configuration parameters for NTP interfaces.";
    list ntp-interface {
        key "ntp-ifname";
        leaf ntp-ifname {
            type leafref {
                path "/if:interfaces/if:interface/if:name";
            }
        }
    }
    container multicast-client {
        leaf multicast-client-address {
            type union {
                type multicast-client-v4address;
                type multicast-client-v6address;
            }
        }
        description
            "The IP address of the multicast group to join.";
    }
}
container multicast-server {
    leaf multicast-server-address {
        type inet:ip-address;
    }
    description
        "The IP address to send NTP multicast packets.";
    }
    leaf multicast-server-ttl {
        type uint8;
    }
    description
        "Specifies the time to live (TTL) of a multicast packet.";
    }
    leaf multicast-server-version {
        type ntp-version;
    }
    description
```



```
        "Specifies the version a multicast packet.";
    }
    leaf multicast-server-keyid {
        type leafref {
            path "/ntp:ntp-cfg/ntp:authentication/ntp:authentication-
keys/ntp:key-id";
        }
        description
            "Specifies the authentication key id of a multicast
packet.";
    }
}
container broadcast-client {
    leaf broadcast-client-enabled {
        type boolean;
        description
            "Allows a device to receive NTP broadcast packets on an
interface.";
    }
}
container broadcast-server {
    leaf broadcast-server-version {
        type ntp-version;
        description
            "Specifies the version of a broadcast packet.";
    }
    leaf broadcast-server-keyid {
        type leafref {
            path "/ntp:ntp-cfg/ntp:authentication/ntp:authentication-
keys/ntp:key-id";
        }
        description
            "Specifies the authentication key id of a broadcast
packet.";
    }
}
}

/* Operational state data */
container ntp-state {
    config "false";
    description
        "Operational state of the NTP.;"
```

```
container system-status {  
  description  
    "System status of NTP.";  
  leaf clock-state {  
    type enumeration {
```

```
        enum synchronized {
            description
                "Indicates that the local clock has been
                synchronized with an NTP server
                or the reference clock.";
        }
        enum unsynchronized {
            description
                "Indicates that the local clock has not been
                synchronized with any NTP server.";
        }
    }
    description
        "Indicates the state of system clock.";
}
leaf clock-stratum {
    type ntp-stratum;
    description
        "Indicates the stratum of the reference clock.";
}
leaf clock-refid {
    type union {
        type inet:ipv4-address;
        type binary {
            length "4";
        }
        type string {
            length "4";
        }
    }
    description
        "IPv4 address or first 32 bits of the MD5 hash
        of the IPv6 address or reference clock of the peer
        to which clock is synchronized.";
}
leaf nominal-freq {
    type decimal64 {
        fraction-digits 4;
    }
    description
        "Indicates the nominal frequency of the local clock, in
        Hz.";
}
leaf actual-freq {
    type decimal64 {
        fraction-digits 4;
    }
    description
```

Hz."; "Indicates the actual frequency of the local clock, in

```
    }
    leaf clock-precision {
        type uint8;
        description
            "Precision of the clock of this system in Hz.(prec=2^(-
n))";
    }
    leaf clock-offset {
        type decimal64 {
            fraction-digits 4;
        }
        description
            "Offset of clock to synchronized peer, in milliseconds.";
    }
    leaf root-delay {
        type decimal64 {
            fraction-digits 2;
        }
        description
            "Total delay along path to root clock, in milliseconds.";
    }
    leaf root-dispersion {
        type decimal64 {
            fraction-digits 2;
        }
        description
            "Indicates the dispersion between the local clock
            and the master reference clock, in milliseconds.";
    }
    leaf peer-dispersion {
        type decimal64 {
            fraction-digits 2;
        }
        description
            "Indicates the dispersion between the local clock
            and the peer clock, in milliseconds.";
    }
    leaf reference-time {
        type string;
        description "Indicates reference timestamp.";
    }
    leaf sync-state {
        type enumeration {
            enum clock-not-set {
                description
                    "Indicates the clock is not updated.";
            }
            enum freq-set-by-cfg {
```

description

Wu & Kumar S N

Expires August 1, 2015

[Page 17]

```
        "Indicates the clock frequency is set
        by NTP configuration.";
    }
    enum clock-set {
        description
            "Indicates the clock is set.";
    }
    enum freq-not-determined {
        description
            "Indicates the clock is set but the frequency
            is not determined.";
    }
    enum clock-synchronized {
        description
            "Indicates that the clock is synchronized.";
    }
    enum spike {
        description
            "Indicates a time difference of more than
            128 milliseconds is detected between NTP server
            and client clock. The clock change will take effect
            in XXX seconds.";
    }
}
description
    "Indicates the synchronization status of the local clock.";
}

container associations-status {
    description
        "System status of NTP.";
    list association-status {
        key "association-source";
        leaf association-source {
            type union {
                type inet:ipv4-address;
                type inet:ipv6-address;
            }
            description
                "IPv4 or IPv6 address of the peer.
                If a nondefault VRF is configured for the peer,
                the VRF follows the address.";
        }
    }
    leaf association-stratum {
        type ntp-stratum;
        description
            "Indicates the stratum of the reference clock.";
    }
}
```



```
    }
    leaf association-refid {
        type union {
            type inet:ipv4-address;
            type binary {
                length "4";
            }
            type string {
                length "4";
            }
        }
        description
            "Reference clock type or address for the peer.";
    }
    leaf association-reach {
        type uint8;
        description
            "Indicates the reachability of the configured server or
peer.";
    }
    leaf association-poll {
        type uint8;
        description
            "Indicates the polling interval for current, in
seconds.";
    }
    leaf association-now {
        type uint32;
        description
            "Indicates the time since the NTP packet was not
            received or last synchronized, in seconds.";
    }
    leaf association-offset {
        type decimal64 {
            fraction-digits 4;
        }
        description
            "Indicates the offset between the local clock
            and the superior reference clock.";
    }
    leaf association-delay {
        type decimal64 {
            fraction-digits 2;
        }
        description
            "Indicates the delay between the local clock
            and the superior reference clock.";
    }
}
```

```
leaf association-dispersion {  
  type decimal64 {
```

```
        fraction-digits 2;
    }
    description
        "Indicates the dispersion between the local clock
        and the superior reference clock.";
}
leaf association-sent {
    type uint32;
    description
        "Indicates the total number of packets
        this association sent.";
}
leaf association-sent-fail {
    type uint32;
    description
        "Indicates the number of times packet sending
        failed by this association.";
}
leaf association-received {
    type uint32;
    description
        "Indicates the total number of packets
        this association received.";
}
leaf association-dropped {
    type uint32;
    description
        "Indicates the number of packets
        this association dropped.";
}
}

container ntp-statistics {
    description
        "Packet statistics of NTP.";
    leaf packet-sent {
        type uint32;
        description
            "Indicates the total number of packets sent.";
    }
    leaf packet-sent-fail {
        type uint32;
        description
            "Indicates the number of times packet sending failed.";
    }
    leaf packet-received {
        type uint32;
```



```
        description
            "Indicates the total number of packets received.";
    }
    leaf packet-dropped {
        type uint32;
        description
            "Indicates the number of packets dropped.";
    }
}

}

}
```

//<CODE ENDS>

## 5. IANA Considerations

This document registers a URI in the "IETF XML Registry" [[RFC3688](#)]. Following the format in [RFC 3688](#), the following registration has been made.

URI: urn:ietf:params:xml:ns:yang:ietf-ntp

Registrant Contact: The NETMOD WG of the IETF.

XML: N/A; the requested URI is an XML namespace.

This document registers a YANG module in the "YANG Module Names" registry [[RFC6020](#)].

Name: ietf-ntp

Namespace: urn:ietf:params:xml:ns:yang:ietf-ntp

Prefix: ntp

Reference: RFC XXXX

## 6. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [[RFC6241](#)]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [[RFC6242](#)]. The NETCONF access control model [[RFC6536](#)] provides the means to restrict access for particular



NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

There are a number of data nodes defined in the YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., <edit-config>) to these data nodes without proper protection can have a negative effect on network operations.

## **7. Acknowledgments**

TBD.

## **8. References**

### **8.1. Normative References**

- [RFC3688] Mealling, M., "The IETF XML Registry", [BCP 81](#), [RFC 3688](#), January 2004.
- [RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", [RFC 5905](#), June 2010.
- [RFC5907] Gerstung, H., Elliott, C., and B. Haberman, "Definitions of Managed Objects for Network Time Protocol Version 4 (NTPv4)", [RFC 5907](#), June 2010.
- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), October 2010.
- [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.



## **8.2. Informative References**

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

### Authors' Addresses

Nan Wu  
Huawei  
Huawei Bld., No.156 Beiqing Rd.  
Beijing 100095  
China

Email: [eric.wu@huawei.com](mailto:eric.wu@huawei.com)

Anil Kumar S N  
Huawei  
Kundalahalli Village, Whitefield  
Bangalore, Kanataka 560037  
India

Email: [anil.sn@huawei.com](mailto:anil.sn@huawei.com)

