

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 13, 2017

N. Wu
Huawei
A. Kumar S N
RtBrick Inc.
Y. Zhao
Ericsson
D. Dhody
A. Sinha
Huawei
May 12, 2017

A YANG Data Model for NTP
draft-wu-ntp-ntp-cfg-03

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 November 13, 2017.

Copyright Notice

Copyright (c) 2017 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

1. Introduction	2
1.1. Operational State	3
1.2. Terminology	3
1.3. Tree Diagrams	3
2. NTP data model	3
3. Relationship with NTPv4-MIB	6
4. Relationship with RFC7317	7
5. NTP YANG Module	8
6. IANA Considerations	25
7. Security Considerations	26
8. Acknowledgments	26
9. References	26
9.1. Normative References	26
9.2. Informative References	27
Authors' Addresses	28

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. Operational State

NTP Operational State is included in the same tree as NTP configuration, consistent with Network Management Datastore Architecture [I-D.ietf-netmod-revised-datastores]. NTP current state and statistics are also maintained in the operational state. Additionally, the operational state also include the associations state.

1.2. Terminology

1.3. 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!
  |   +--rw port?          uint16
  |   +--rw refclock-master!
  |   |   +--rw master-stratum?  ntp-stratum
  |   +--rw authentication
  |   |   +--rw auth-enabled?    boolean
  |   |   +--rw trusted-keys* [key-id]
  |   |   |   +--rw key-id      leafref
  |   |   +--rw authentication-keys* [key-id]
  |   |   |   +--rw key-id      uint32
  |   |   +--rw algorithm?     identityref
```

```

|      +--rw password?      ianach:crypt-hash
+--rw access-rules
|      +--rw access-rule* [access-mode]
|      |      +--rw access-mode      access-modes
|      |      +--rw acl?              -> /acl:access-lists/acl/acl-name
+--ro clock-state
|      +--ro system-status
|      |      +--ro clock-state?      ntp-clock-status
|      |      +--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?   yang:date-and-time
|      |      +--ro sync-state?      ntp-sync-state
+--rw associations* [address association-type]
|      +--rw address            inet:host
|      +--rw association-type    association-modes
|      +--rw authentication
|      |      +--rw (authentication-type)?
|      |      |      +---:(symmetric-key)
|      |      |      |      +--rw key-id?  leafref
+--rw prefer?                  boolean
+--rw burst?                   boolean
+--rw iburst?                  boolean
+--rw source?                  if:interface-ref
+--rw minpoll?                 ntp-minpoll
+--rw maxpoll?                 ntp-maxpoll
+--rw port?                    uint16
+--rw version?                 ntp-version
+--ro stratum?                 ntp-stratum
+--ro refid?                   union
+--ro reach?                   uint8
+--ro unreach?                 uint8
+--ro poll?                    uint8
+--ro now?                     uint32
+--ro offset?                  decimal64
+--ro delay?                   decimal64
+--ro dispersion?              decimal64
+--ro originate-time?          yang:date-and-time
+--ro receive-time?            yang:date-and-time
+--ro transmit-time?           yang:date-and-time
+--ro input-time?              yang:date-and-time
+--ro ntp-statistics

```

```

|      +--ro packet-sent?          yang:counter32
|      +--ro packet-sent-fail?     yang:counter32
|      +--ro packet-received?      yang:counter32
|      +--ro packet-dropped?       yang:counter32
+--rw interface
  +--rw interface* [interface]
    +--rw interface                if:interface-ref
    +--rw broadcast-server!
      +--rw ttl?                   uint8
      +--rw authentication
        +--rw (authentication-type)?
          +--:(symmetric-key)
            +--rw key-id?          leafref
      +--rw minpoll?               ntp-minpoll
      +--rw maxpoll?               ntp-maxpoll
      +--rw port?                  uint16
      +--rw version?               ntp-version
    +--rw broadcast-client!
    +--rw multicast-server* [address]
      +--rw address
        | rt-types:ip-multicast-group-address
      +--rw ttl?                   uint8
      +--rw authentication
        +--rw (authentication-type)?
          +--:(symmetric-key)
            +--rw key-id?          leafref
      +--rw minpoll?               ntp-minpoll
      +--rw maxpoll?               ntp-maxpoll
      +--rw port?                  uint16
      +--rw version?               ntp-version
    +--rw multicast-client* [address]
      +--rw address                rt-types:ip-multicast-group-address
    +--rw manycast-server* [address]
      +--rw address                rt-types:ip-multicast-group-address
    +--rw manycast-client* [address]
      +--rw address
        | rt-types:ip-multicast-group-address
      +--rw authentication
        +--rw (authentication-type)?
          +--:(symmetric-key)
            +--rw key-id?          leafref
      +--rw ttl?                   uint8
      +--rw minclock?              uint8
      +--rw maxclock?              uint8
      +--rw beacon?                uint8
      +--rw minpoll?               ntp-minpoll
      +--rw maxpoll?               ntp-maxpoll
      +--rw port?                  uint16

```

```

|           +--rw version?           ntp-version
+--ro ntp-statistics
  +--ro packet-sent?                 yang:counter32
  +--ro packet-sent-fail?            yang:counter32
  +--ro packet-received?             yang:counter32
  +--ro packet-dropped?              yang:counter32

```

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 with NTPv4-MIB

If the device implements the NTPv4-MIB [RFC5907], data nodes in container ntp 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/	NTPv4-MIB objects
ntp-enabled	ntpEntStatusCurrentMode

YANG data nodes in /ntp/associations	NTPv4-MIB objects
address	ntpAssocAddressType ntpAssocAddress

YANG NTP Configuration Data Nodes and Related NTPv4-MIB Objects

YANG data nodes in /ntp-state /system-status	NTPv4-MIB objects
clock-state clock-stratum clock-refid clock-precision clock-offset root-dispersion	ntpEntStatusCurrentMode ntpEntStatusStratum ntpEntStatusActiveRefSourceId ntpEntStatusActiveRefSourceName ntpEntTimePrecision ntpEntStatusActiveOffset ntpEntStatusDispersion
YANG data nodes in /ntp-state /associations-status/association-status/	NTPv4-MIB objects
association-source association-stratum association-refid association-offset association-delay association-dispersion association-sent association-received association-dropped	ntpAssocAddressType ntpAssocAddress ntpAssocStratum ntpAssocRefId ntpAssocOffset ntpAssocStatusDelay ntpAssocStatusDispersion ntpAssocStatOutPkts ntpAssocStatInPkts ntpAssocStatProtocolError

YANG NTP State Data Nodes and Related NTPv4-MIB Objects

4. Relationship with RFC7317

This section describes the relationship with NTP definition in Section 3.2 System Time Management of [RFC7317] . YANG data nodes in /ntp/ also supports interface related configurations which is not supported in /system/ntp

YANG data nodes in /ntp/	YANG data nodes in /system/ntp
ntp-enabled	enabled
associations/association	server
	server/name
associations/association/address	server/transport/udp/address
ntp-enabled/port	server/transport/udp/port
associations/association-type	server/association-type
associations/association/iburst	server/iburst
associations/association/prefer	server/prefer

YANG NTP Configuration Data Nodes and counterparts in RFC7317 Objects

5. NTP YANG Module

```
<CODE BEGINS> file "ietf-ntp@2017-05-12.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 iana-crypt-hash {
        prefix ianach;
    }

    import ietf-key-chain {
        prefix "key-chain";
    }

    import ietf-access-control-list {
        prefix "acl";
    }
}
```



```
import ietf-routing-types {
  prefix "rt-types";
}

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>
  Editor:   Anil Kumar S N
           <mailto:anil.ietf@gmail.com>
  Editor:   Yi Zhao
           <mailto:yi.z.zhao@ericsson.com>
  Editor:   Dhruv Dhody
           <mailto:dhruv.ietf@gmail.com>
  Editor:   Ankit Kumar Sinha
           <mailto:ankit.ietf@gmail.com>";

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

  Copyright (c) 2017 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 2017-05-12 {
  description
    "Updated revision.";
  reference
    "RFC XXXX: A YANG Data Model for NTP Management";
}

/* Typedef Definitions */
```

```
typedef ntp-stratum {
  type uint8 {
    range "1..16";
  }
  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 minimum poll exponent for this NTP association.";
}

typedef ntp-maxpoll {
  type uint8 {
    range "4..17";
  }
  default "10";
  description
    "The maximul poll exponent for this NTP association.";
}

typedef access-modes {
  type enumeration {
    enum peer {
      value "0";
      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 {
        value "1";
        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 {
        value "2";
        description
            "Enables the server to access.
            Only time request can be performed on the
            local NTP service.";
    }
    enum query {
        value "3";
        description
            "Sets the maximum access limitation.
            Control query can be performed only on the
            local NTP service.";
    }
}
description
    "This defines NTP access modes.";
}

typedef association-modes {
    type enumeration {
        enum server {
            value "0";
            description
                "Use client association mode. This device
                will not provide synchronization to the
                configured NTP server.";
        }
        enum peer {
            value "1";
            description
                "Use symmetric active association mode.
                This device may provide synchronization
                to the configured NTP server.";
        }
        enum pool {
            value "2";
```

```
        description
            "Use client association mode with one or
            more of the NTP servers found by DNS
            resolution of the domain name given by
            the 'address' leaf. This device will not
            provide synchronization to the servers.";
    }
}
description
    "This defines NTP association modes.";
}

typedef ntp-clock-status {
    type enumeration {
        enum synchronized {
            value "0";
            description
                "Indicates that the local clock has been
                synchronized with an NTP server or
                the reference clock.";
        }
        enum unsynchronized {
            value "1";
            description
                "Indicates that the local clock has not been
                synchronized with any NTP server.";
        }
    }
}
description
    "This defines NTP clock status.";
}

typedef ntp-sync-state {
    type enumeration {
        enum clock-not-set {
            value "0";
            description
                "Indicates the clock is not updated.";
        }
        enum freq-set-by-cfg {
            value "1";
            description
                "Indicates the clock frequency is set by
                NTP configuration.";
        }
        enum clock-set {
            value "2";
            description
```

```
        "Indicates the clock is set.";
    }
    enum freq-not-determined {
        value "3";
        description
            "Indicates the clock is set but the frequency
             is not determined.";
    }
    enum clock-synchronized {
        value "4";
        description
            "Indicates that the clock is synchronized";
    }
    enum spike {
        value "5";
        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
    "This defines NTP clock sync states.";
}

/* Groupings */
grouping authentication-key {
    description
        "To define an authentication key for a Network Time
         Protocol (NTP) time source.";
    leaf key-id {
        type uint32 {
            range "1..max";
        }
        description
            "Authentication key identifier.";
    }
    leaf algorithm {
        type identityref {
            base key-chain:crypto-algorithm;
        }
        description
            "Authentication algorithm.";
    }
    leaf password {
        type ianach:crypt-hash;
        description "Clear or encrypted mode for password text.";
    }
}
```

```
    }  
  }  
  
  grouping authentication-type-param {  
    description  
      "Authentication type.";  
    choice authentication-type {  
      description  
        "Type of authentication.";  
      case symmetric-key {  
        leaf key-id {  
          type leafref {  
            path "/ntp:ntp/ntp:authentication/"  
              + "ntp:authentication-keys/ntp:key-id";  
          }  
          description  
            "Authentication key id referenced in this  
              association.";  
        }  
      }  
    }  
  }  
}  
  
grouping statistics {  
  description  
    "NTP packet statistic.";  
  leaf packet-sent {  
    type yang:counter32;  
    description  
      "Indicates the total number of packets sent.";  
  }  
  leaf packet-sent-fail {  
    type yang:counter32;  
    description  
      "Indicates the number of times packet  
        sending failed.";  
  }  
  leaf packet-received {  
    type yang:counter32;  
    description  
      "Indicates the total number of packets received.";  
  }  
  leaf packet-dropped {  
    type yang:counter32;  
    description  
      "Indicates the number of packets dropped.";  
  }  
}
```

```
grouping common-attributes {
  description
    "NTP common attributes for configuration.";
  leaf minpoll {
    type ntp-minpoll;
    description
      "The minimum poll interval used in this association.";
  }
  leaf maxpoll {
    type ntp-maxpoll;
    description
      "The maximum poll interval used in this association.";
  }
  leaf port {
    type uint16 {
      range "123 | 1025..max";
    }
    default "123";
    description
      "Specify the port used to send NTP packets.";
  }
  leaf version {
    type ntp-version;
    description
      "NTP version.";
  }
}

/* Configuration data nodes */
container ntp {
  presence
    "NTP is enable";
  description
    "Configuration parameters for NTP.";
  leaf port {
    type uint16 {
      range "123 | 1025..max";
    }
    default "123";
    description
      "Specify the port used to send NTP packets.";
  }
}

container refclock-master {
  presence
    "NTP master clock is enable";
  description
    "Configures the device as NTP server.";
```

```
    leaf master-stratum {
        type ntp-stratum;
        default "16";
        description
            "Stratum level from which NTP
             clients get their time synchronized.";
    }
}
container authentication {
    description
        "Configuration of authentication.";
    leaf auth-enabled {
        type boolean;
        default false;
        description
            "Controls whether NTP authentication is enabled
             or disabled on this device.";
    }
    list trusted-keys {
        key "key-id";
        description
            "List of keys trusted by NTP.";
        leaf key-id {
            type leafref {
                path "/ntp:ntp/ntp:authentication/"
                    + "ntp:authentication-keys/ntp:key-id";
            }
            description
                "The key trusted by NTP.";
        }
    }
    list authentication-keys {
        key "key-id";
        uses authentication-key;
        description
            "List of authentication key.";
    }
}

container access-rules {
    description
        "Configuration of access rules.";
    list access-rule {
        key "access-mode";
        description
            "List of access rules.";
        leaf access-mode {
            type access-modes;
        }
    }
}
```



```
        description
            "NTP access mode.";
    }
    leaf acl {
        type leafref {
            path "/acl:access-lists/acl:acl/acl:acl-name";
        }
        description
            "NTP ACL.";
    }
}

container clock-state {
    config "false";
    description
        "Operational state of the NTP.";

    container system-status {
        description
            "System status of NTP.";
        leaf clock-state {
            type ntp-clock-status;
            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
      "Indicates the actual frequency of the
       local clock, in Hz.";
  }
  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 yang:date-and-time;
        description
            "Indicates reference timestamp.";
    }
    leaf sync-state {
        type ntp-sync-state;
        description
            "Indicates the synchronization status of
            the local clock.";
    }
}

list associations {
    key "address association-type";
    description
        "list of association.";
    leaf address {
        type inet:host;
        description
            "The address of this association.";
    }
    leaf association-type {
        type association-modes;
        description
            "The desired association type for this NTP server.";
    }
    container authentication{
        description
            "Authentication type.";
        uses authentication-type-param;
    }
    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 source {
        type if:interface-ref;
        description
            "The interface whose ip address this association used
            as source address.";
    }
    uses common-attributes {
        description
            "Common attribute like port, version, min and max poll.";
    }
    leaf stratum {
        type ntp-stratum;
        config "false";
        description
            "Indicates the stratum of the reference clock.";
    }
    leaf refid {
        type union {
            type inet:ipv4-address;
            type binary {
                length "4";
            }
            type string {
                length "4";
            }
        }
        config "false";
        description
            "Reference clock type or address for the peer.";
    }
    leaf reach {
        type uint8;
        config "false";
        description
            "Indicates the reachability of the configured
            server or peer.";
    }
    leaf unreach {
```

```
    type uint8;
    config "false";
    description
        "Indicates the unreachability of the configured
        server or peer.";
}
leaf poll {
    type uint8;
    config "false";
    description
        "Indicates the polling interval for current,
        in seconds.";
}
leaf now {
    type uint32;
    config "false";
    description
        "Indicates the time since the NTP packet was
        not received or last synchronized, in seconds.";
}
leaf offset {
    type decimal64 {
        fraction-digits 4;
    }
    config "false";
    description
        "Indicates the offset between the local clock
        and the superior reference clock.";
}
leaf delay {
    type decimal64 {
        fraction-digits 2;
    }
    config "false";
    description
        "Indicates the delay between the local clock
        and the superior reference clock.";
}
leaf dispersion {
    type decimal64 {
        fraction-digits 2;
    }
    config "false";
    description
        "Indicates the dispersion between the local
        clock and the superior reference clock.";
}
leaf originate-time {
```

```
    type yang:date-and-time;
    config "false";
    description
        "Indicates packet originate timestamp(T1).";
}
leaf receive-time {
    type yang:date-and-time;
    config "false";
    description
        "Indicates packet receive timestamp(T2).";
}
leaf transmit-time {
    type yang:date-and-time;
    config "false";
    description
        "Indicates packet transmit timestamp(T3).";
}
leaf input-time {
    type yang:date-and-time;
    config "false";
    description
        "Indicates packet input timestamp(T4).";
}
container ntp-statistics {
    config "false";
    description
        "Per Peer packet send and receive statistic.";
    uses statistics {
        description
            "NTP send and receive packet statistic.";
    }
}

container interface {
    description
        "Configuration parameters for NTP interfaces.";
    list interface {
        key "interface";
        description
            "List of interfaces.";
        leaf interface {
            type if:interface-ref;
            description
                "The interface name.";
        }

        container broadcast-server {
```

```
presence
  "NTP broadcast-server is configured";
description
  "Configuration of broadcast server.";
leaf ttl {
  type uint8;
  description
    "Specifies the time to live (TTL) of a
    broadcast packet.";
}
container authentication{
  description
    "Authentication type.";
  uses authentication-type-param;
}
uses common-attributes {
  description
    "Common attribute like port, version, min and max poll.";
}
}

container broadcast-client {
  presence
    "NTP broadcast-client is configured";
  description
    "Configuration of broadcast-client.";
}

list multicast-server {
  key "address";
  description
    "Configuration of multicast server.";
  leaf address {
    type rt-types:ip-multicast-group-address;
    description
      "The IP address to send NTP multicast packets.";
  }
  leaf ttl {
    type uint8;
    description
      "Specifies the time to live (TTL) of a
      multicast packet.";
  }
  container authentication{
    description
      "Authentication type.";
    uses authentication-type-param;
  }
}
```

```
    uses common-attributes {
      description
        "Common attribute like port, version, min and max poll.";
    }
  }
  list multicast-client {
    key "address";
    description
      "Configuration of multicast-client.";
    leaf address {
      type rt-types:ip-multicast-group-address;
      description
        "The IP address of the multicast group to join.";
    }
  }
  list manycast-server {
    key "address";
    description
      "Configuration of manycast server.";
    leaf address {
      type rt-types:ip-multicast-group-address;
      description
        "The multicast group IP address to receive
        manycast client messages .";
    }
  }
  list manycast-client {
    key "address";
    description
      "Configuration of manycast-client.";
    leaf address {
      type rt-types:ip-multicast-group-address;
      description
        "The group IP address that the manycast client
        broadcasts the request message to.";
    }
  }
  container authentication{
    description
      "Authentication type.";
    uses authentication-type-param;
  }
  leaf ttl {
    type uint8;
    description
      "Specifies the maximum time to live (TTL) for
      the expanding ring search.";
  }
  leaf minclock {
```



```

        type uint8;
        description
            "The minimum manycast survivors in this
            association.";
    }
    leaf maxclock {
        type uint8;
        description
            "The maximum manycast candidates in this
            association.";
    }
    leaf beacon {
        type uint8;
        description
            "The maximum interval between beacons in this
            association.";
    }
    uses common-attributes {
        description
            "Common attribute like port, version, min and max poll.";
    }
}
}
}
}

/* Operational state data */

container ntp-statistics {
    config "false";
    description
        "Total NTP packet statistic.";
    uses statistics {
        description
            "NTP send and receive packet statistic.";
    }
}
}
}
<CODE ENDS>

```

6. 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

7. 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.

8. Acknowledgments

The authors would like to express their thanks to Sladjana Zoric, Danny Mayer, Harlan Stenn, Ulrich Windl and Miroslav Lichvar for their review and suggestions.

9. References

9.1. Normative References

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

- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <<http://www.rfc-editor.org/info/rfc5905>>.
- [RFC5907] Gerstung, H., Elliott, C., and B. Haberman, Ed., "Definitions of Managed Objects for Network Time Protocol Version 4 (NTPv4)", RFC 5907, DOI 10.17487/RFC5907, June 2010, <<http://www.rfc-editor.org/info/rfc5907>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<http://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, <<http://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<http://www.rfc-editor.org/info/rfc6242>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<http://www.rfc-editor.org/info/rfc6536>>.

9.2. Informative References

- [I-D.ietf-netmod-revised-datastores] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture", draft-ietf-netmod-revised-datastores-02 (work in progress), May 2017.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC7317] Bierman, A. and M. Bjorklund, "A YANG Data Model for System Management", RFC 7317, DOI 10.17487/RFC7317, August 2014, <<http://www.rfc-editor.org/info/rfc7317>>.

Authors' Addresses

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

Email: eric.wu@huawei.com

Anil Kumar S N
RtBrick Inc.
Bangalore, Kanataka 560037
India

Email: anil.ietf@gmail.com

Yi Zhao
Ericsson
China Digital Kingdom Bld., No.1 WangJing North Rd.
Beijing 100102
China

Email: yi.z.zhao@ericsson.com

Dhruv Dhody
Huawei
Divyashree Techno Park, Whitefield
Bangalore, Kanataka 560066
India

Email: dhruv.ietf@gmail.com

Ankit kumar Sinha
Huawei
Divyashree Techno Park, Whitefield
Bangalore, Kanataka 560066
India

Email: ankit.ietf@gmail.com