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 June 29, 2017.

Copyright Notice

Copyright (c) 2016 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
1. Introduction


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]:

- configuration data
- data model
- module
- 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:

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

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 port?         uint16
      |    +--rw refclock-master
      |         +--rw master?    boolean
      |         +--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? enumeration
         |         |    +--rw password? ianach:crypt-hash
         +--rw autokey
            |    +--rw automax?     uint8
            |    +--rw crypto?      enumeration
            |         |    +--rw digest?     enumeration
            +--rw ident?         string
```
Internet-Draft                YANG for NTP                 December 2016

| ++--rw revoke?      uint8
++--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 association* [address vrf]
    ++--rw version?    ntp-version
    ++--rw address     inet:host
    ++--rw association-type?    enumeration
    ++--rw authentication
      |   ++--rw (authentication-type)?
      |      ++--:(symmetric-key)
      |      |   ++--rw key-id?    leafref
      |      |      ++--:(auto-key)
      |      |   ++--rw auto-key?    empty
    ++--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 broadcast-multicast-server
      |   ++--rw address?    inet:ip-address
      |   ++--rw ttl?        uint8
      |   ++--rw version?    ntp-version
      |   ++--rw authentication
      |      |   ++--rw (authentication-type)?
      |      |      ++--:(symmetric-key)
      |      |      |   ++--rw key-id?    leafref
      |      |      |      ++--:(auto-key)
      |      |      |   ++--rw auto-key?    empty
      |      |   ++--rw minpoll?   ntp-minpoll
    ++--rw broadcast-client
      |   ++--rw broadcast-client-enabled?    boolean
    ++--rw multicast-client
      |   ++--rw multicast-client-address?    union
    ++--rw manycast-server
      |   ++--rw manycast-server-address?    inet:ip-address
    ++--rw manycast-client
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-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.

<table>
<thead>
<tr>
<th>YANG data nodes in /ntp-cfg/</th>
<th>NTPv4-MIB objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>ntp-enabled</td>
<td>ntpEntStatusCurrentMode</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>YANG data nodes in /ntp-cfg/associations</th>
<th>NTPv4-MIB objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>ntpAssocAddressType</td>
</tr>
<tr>
<td></td>
<td>ntpAssocAddress</td>
</tr>
</tbody>
</table>

YANG NTP Configuration 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-cfg/                          | YANG data nodes in /system/ntp |
+-----------------------------------------------------+-------------------------------+
| ntp-enabled                                          | enabled                       |
| associations/association                             | server                        |
| associations/association/address                     | server/name                   |
| ntp-enabled/port                                     | server/transport/udp/address   |
| associations/association-type                        | server/transport/udp/port      |
| ntpinterfaces/broadcast-multicast-server             | server/association-type       |
| ntpinterfaces/broadcast-client                       |                               |
| ntpinterfaces/multicast-server                       |                               |
| ntpinterfaces/manycast-server                        |                               |
| ntpinterfaces/manycast-client                        |                               |
| 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@2016-11-01.yang"
module ietf-ntp {
    namespace "urn:ietf:params:xml:ns:yang:ietf-ntp";
    prefix "ntp";
    import ietf-inet-types {
        prefix "inet";
    }
    import ietf-interfaces {
        prefix "if";
    }
    import iana-crypt-hash {
        prefix ianach;
    }
    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>

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 "2016-11-01" {
  description "Second revision " +
    " - changed trusted-keys to list " +
    " - changed password to type ianach:crypt-hash " +
    " - rearranged peer and server " +
    " - added association type pool " +
    " - mergerd multicast server and broadcast server " +
    " - added auto-key authentication type " +
    " - added container manycast server" +
    " - added container manycast client" +
    " - added a new leaf association-unreach";
  reference "";
}

revision 2015-09-27 {
  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 maximum poll exponent 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 Network Time Protocol (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 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-cfg/ntp:authentication/
 + "ntp:authentication-keys/ntp:key-id";
}
description
"Authentication key id referenced in this association."
);
}
case auto-key {
  leaf auto-key {
    type empty;
    description
    "Autokey authentication."
  };
}
/* 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."
  }
leaf port {
  type uint16 {
    range "123 | 1025..max";
  }
  default "123";
  description
  "Specify the port used to send NTP packets."
}
container refclock-master {
  description
  "Configuration for reference clock.";
  leaf master {
    type boolean;
    default false;
    description
    "Use its own NTP master clock to synchronize with peers
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.";
    }
    list trusted-keys {
        key "key-id";
        description "List of keys trusted by NTP.";
        leaf key-id {
            type leafref {
                path "/ntp:ntp-cfg/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 autokey {
        description "Configuration parameters for Autokey.";
        leaf automax {
            type uint8;
            default "12";
        }
    }
}

description "The interval between regenerations of the session key list used with the Autokey protocol, as a power of 2 in seconds.";

container crypto {
    description "Configuration parameters for Autokey public key cryptography";
    leaf digest {
        type enumeration {
            enum md2 {
                description "Message Digest 2 (MD2) algorithm.";
            }
            enum md4 {
                description "Message Digest 4 (MD4) algorithm.";
            }
            enum md5 {
                description "Message Digest 5 (MD5) algorithm.";
            }
            enum mdc2 {
                description "Modification Detection Code 2(MDC2) algorithm.";
            }
            enum ripemd160 {
                description "160-bit version of RACE Integrity Primitives Evaluation Message Digest (RIPEMD-160) algorithm.";
            }
            enum sha {
                description "Secure Hash Algorithm (SHA) algorithm.";
            }
            enum sha1 {
                description "Secure Hash Algorithm (SHA1) algorithm.";
            }
        }
        default "md5";
        description "Message digest algorithm.";
    }
}

container access-rules {
    description "Configuration of access rules.";
}
list access-rule {
  key "access-mode";
  description
    "List of access rules.";
  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 {
  description
    "Configuration of acl numbers.";
  choice acl-type {
    description
      "Type of acl.";
    case ipv4 {
      leaf acl-number-ipv4 {
        type uint16;
        description "IPv4 acl number.";
      }
    }
  }
}
case ipv6 {
    leaf acl-number-ipv6 {
        type uint16;
        description "IPv6 acl number."
    }
}

list associations {
    key "address vrf";
    description "list of association."
    leaf version {
        type ntp-version;
        description "NTP version."
    }
    leaf address {
        type inet:host;
        description "The address of this association."
    }
    leaf association-type {
        type enumeration {
            enum server {
                description "Use client association mode. This device will not provide synchronization to the configured NTP server."
            }
            enum peer {
                description "Use symmetric active association mode. This device may provide synchronization to the configured NTP server."
            }
            enum pool {
                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."
            }
        }
    }
}
default server;
description
  "The desired association type for this NTP server.";
}
container authentication{
  description
  "Authentication type.";
  uses authentication-type-param;
}
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 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.";
}
}

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

container broadcast-multicast-server {
    description  
"Configuration of broadcast or multicast server.";
    leaf address {
        type inet:ip-address;
        description  
"The IP address to send NTP broadcast or multicast packets.";
    }
    leaf ttl {
        type uint8;
        description  
"Specifies the time to live (TTL) of a broadcast or multicast packet.";
    }
    leaf version {
        type ntp-version;
        description  
"Specifies the version a multicast or broadcast packet.";
    }
    container authentication{
        description  
"Authentication type.";
        uses authentication-type-param;
    }
}
leaf minpoll {
  type ntp-minpoll;
  description
    "The minimum poll interval for NTP messages.";
}
}

container broadcast-client {
  description
    "Configuration of broadcast-client.";
  leaf broadcast-client-enabled {
    type boolean;
    description
      "Allows a device to receive Network Time Protocol (NTP) broadcast packets on an interface.";
  }
}

container multicast-client {
  description
    "Configuration of 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 manycast-server {
  description
    "Configuration of manycast server.";
  leaf address {
    type inet:ip-address;
    description
      "The multicast group IP address to receive manycast client messages.";
  }
}

container manycast-client {
  description
    "Configuration of manycast-client.";
  leaf address {
    type inet:ip-address;
    description
      "The group IP address that the manycast client broadcasts the request message to.";
  }
  leaf version {
    type ntp-version;
    description
  }

"NTP version."
}
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 minpoll {
  type ntp-minpoll;
  description  "The minimum poll interval for NTP messages."
}
leaf maxpoll {
  type ntp-maxpoll;
  description  "The maximum poll interval for NTP messages."
}
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."
}

/* 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
    "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 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
      "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";
    description
    "List of association status.";
    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-unreach {
  type uint8;
  description
    "Indicates the unreachability 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
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
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.";
  }
}

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.


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


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


9.2. Informative References


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