

# A YANG Data Model for NTP

## draft-wu-ntp-ntp-cfg-00

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

Anil Kumar S N([anil.sn@huawei.com](mailto:anil.sn@huawei.com))

IETF92, Dallas

# Background

## □ Goal

- To define a standardized YANG data model of Network Time Protocol for configuration, state data, RPC and notification.

## □ Way to do that:

- Collect comments from WG.
- Organize the work from interested parties and maybe attend interim meetings.
- Maybe setup a common GIT repository and verify it.

## □ RFC Covered


- RFC5905
- RFC5907

# Overview(Configuration)


## □ NTP YANG tree: ntp-cfg

```
+--rw ntp-cfg!  
| +--rw ntp-enabled?          boolean  
| +--rw refclock-master  
| | +--rw master?           boolean  
| | +--rw master-stratum?   ntp-stratum  
| +--rw authentication!  
|   ... ..  
| +--rw access-rules  
|   ... ..  
| +--rw associations  
|   ... ..  
| +--rw ntp-interfaces  
|   ... ..
```


NTP configuration is modeled as a container.



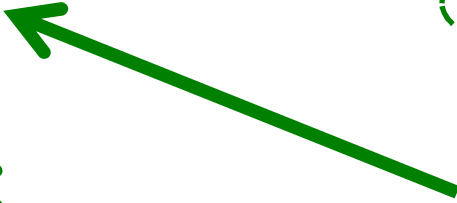
NTP configuration for authentication.



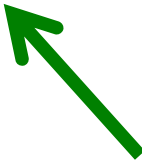
NTP configuration for access rules.



NTP configuration for associations.



NTP configuration for interfaces.



# Overview(Configuration cont.)

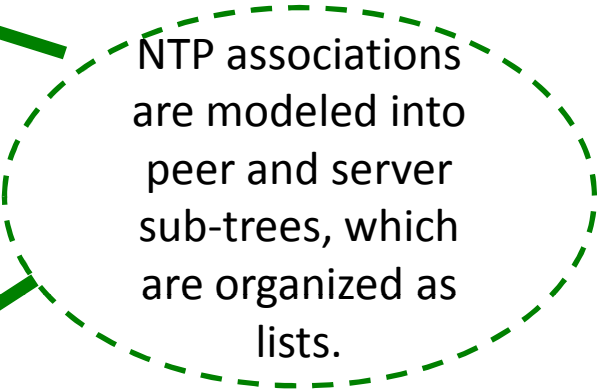
## ▣ NTP YANG tree: ntp-cfg/authentication, ntp-cfg/access-rules

```
|  +--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
```

# Overview(Configuration cont.)

## □ NTP YANG tree: ntp-cfg/associations

```
| +--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
```

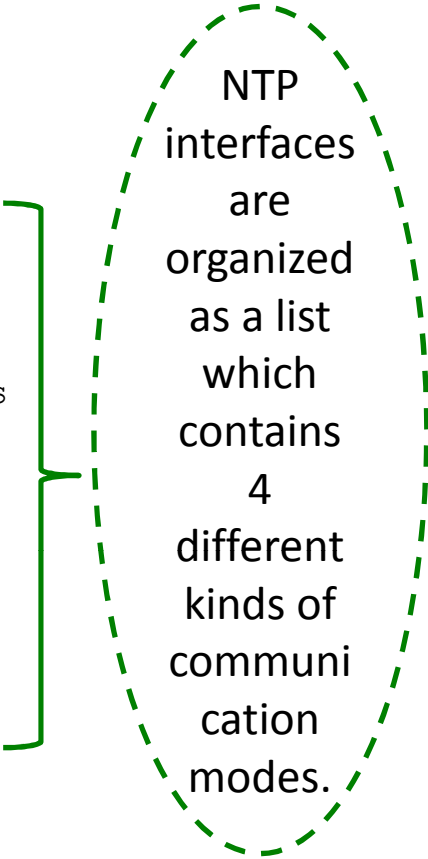


NTP associations are modeled into peer and server sub-trees, which are organized as lists.

# Overview(Configuration cont.)

## □ NTP YANG tree: ntp-cfg/associations

```
| +--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
```

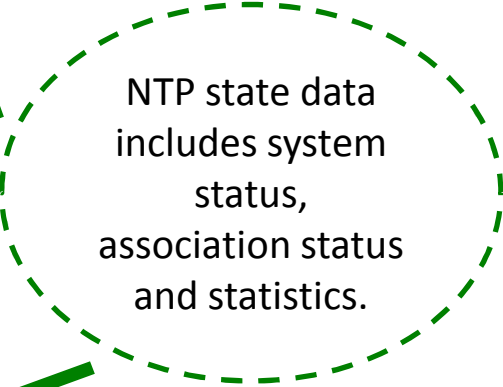


NTP interfaces are organized as a list which contains 4 different kinds of communication modes.

# Overview(State data)

## □ NTP YANG tree: ntp-state

```
module: ietf-ntp
  +--rw ntp-cfg!
    ... ..
  +--ro ntp-state
    +--ro system-status
      ... ..
    +--ro associations-status
      ... ..
    +--ro ntp-statistics
      ... ..
```



NTP state data  
includes system  
status,  
association status  
and statistics.

# Overview(State data cont.)

## □ NTP YANG tree: ntp-state

```
+--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
```



# Overview(State data cont.)

## □ NTP YANG tree: ntp-state

```
+--ro ntp-state
  +--ro system-status
    ... ..
  +--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
```

## Next step

- ❑ Collect feedback and comments for proposed draft.
- ❑ Complete this data model and add missing parts.
  - RPCs
  - Notification
  - Cover more RFCs for flow specification updates
  - ?
- ❑ Looking for co-authors and wish more people to collaborate.