- Network Time Protocol -

Network Time Protocol (NTP)

Time and date information can be configured locally on both Cisco routers and switches. However, in environments with a large amount of equipment, this can become unmanageable. As a result, the "time" will be inconsistent throughout the network. This can lead to various inaccuracies, such as the timestamps in syslog messages.

Network Time Protocol (NTP) provides a mechanism to synchronize time throughout the network. An NTP device will form an **association** with NTP devices closer to the **time source**. NTP devices use a special measurement, called a **stratum**, to determine how far they are away from the time source.

For example, a device with a stratum of 1 is directly connected to the time source. A device with a stratum of 2 is one device (or "hop") away from the time source.

NTP can be configured one of two ways:

- **Client/Server** – The NTP client is configured to always get its time information from the NTP server. The server will never get its time from the client.
- **Peer-to-peer** – Peered NTP devices can get their time from each other, depending on who is closest to the time source (i.e., lowest stratum).
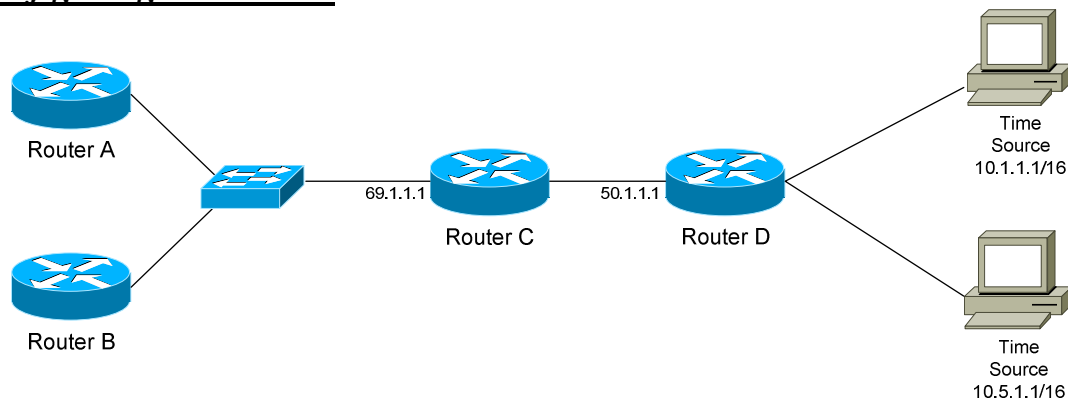
NTP associations can be secured using encrypted authentication.

Configuring Basic NTP

Consider the above example. Two time sources exist off of RouterD; to use those time sources as NTP servers:

> **RouterD(config)#** *clock timezone CST -6*
> **RouterD(config)#** *clock summer-time CDT recurring*
> **RouterD(config)#** *ntp server 10.1.1.1*
> **RouterD(config)#** *ntp server 10.5.1.1*

Router D's time will now be synchronized with the external time sources. Specifying more than one NTP server provides redundancy.

Notice that *timezone* was identified as *CST* (Central Standard Time), which is minus *6* hours off of Greenwich Time. The *clock summer-time* command configures Daily Savings Time (notated as *CDT*, instead of *CST*!)

To *prefer* one time source over another:

> **RouterD(config)#** *ntp server 10.1.1.1 prefer*
> **RouterD(config)#** *ntp server 10.5.1.1*

NTP can be configured as a **tiered** client/server model. For example, instead of pointing RouterC to the external time sources, RouterC can utilize RouterD as its NTP Server instead:
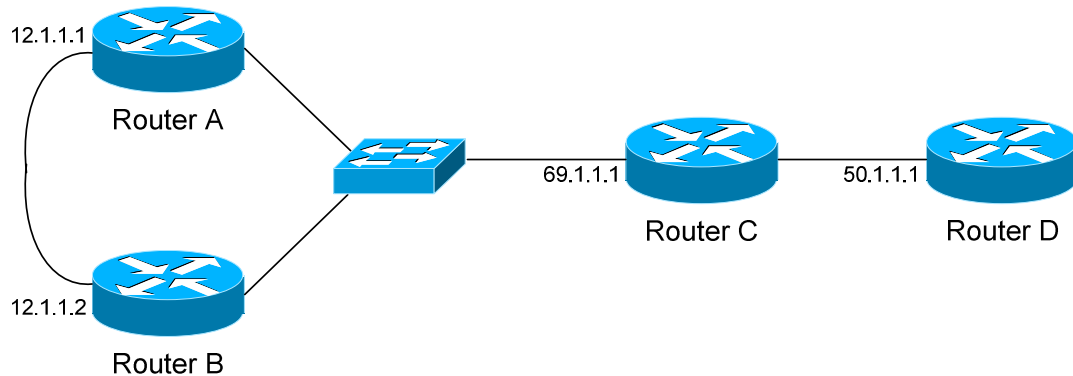
> **RouterC(config)#** *clock timezone CST -6*
> **RouterC(config)#** *clock summer-time CDT recurring*
> **RouterC(config)#** *ntp server 50.1.1.1*

RouterA and RouterB can be pointed to RouterC as well:

> **RouterA(config)#** *clock timezone CST -6*
> **RouterA(config)#** *clock summer-time CDT recurring*
> **RouterA(config)#** *ntp server 69.1.1.1*

Configuring Basic NTP (continued)

Now consider the above example. No external time sources exist; instead, RouterD can be designated as the **authoritative** time source of this network. This is not an optimal solution, and should only be used when there is *no* access to external (Internet) time sources.

First, accurate date and time information must be configured on RouterD:

> **RouterD(config)#** *clock timezone CST -6*
> **RouterD(config)#** *clock summer-time CDT recurring*
>
> **RouterD#** *clock set 04:33:01 June 27 2005*

Next, RouterD must be configured as the NTP *master* time source:

> **RouterD(config)#** *ntp master 10*

The *10* in the *ntp master* command indicates the stratum. Best practice is to set this to a high number, so that it will become less preferred if an external time source becomes available.

Now consider RouterA and RouterB. Both routers should point to RouterC as their NTP Server. However, Router A will lose its direct connection to Router C if its Ethernet interface fails. RouterB could similarly lose its direct connection.
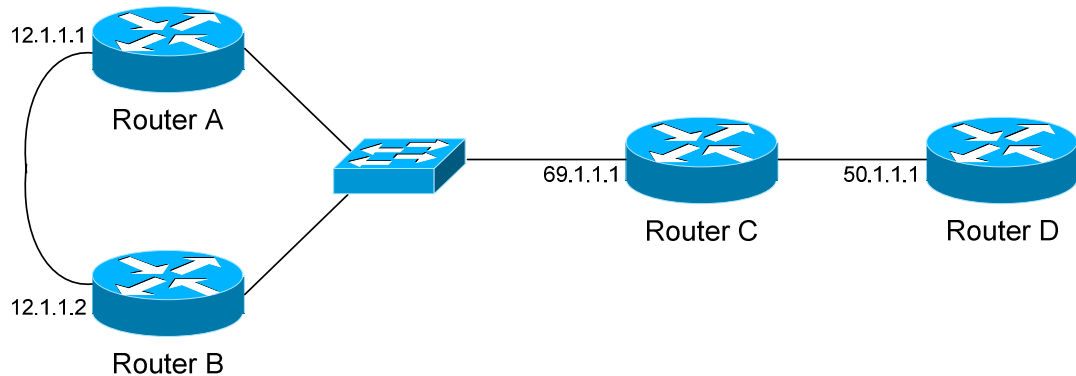
RouterA and RouterB can be configured as "peers," so that one can synchronize the other if either lose their direct connection to the NTP server:

> **RouterA(config)#** *ntp peer 12.1.1.2*
>
> **RouterB(config)#** *ntp peer 12.1.1.1*

NTP Authentication

To ensure that erroneous time information is not propagated throughout the network, NTP authentication can be configured.

Assume that RouterD is still configured as the NTP *master*. To configure authentication on RouterD:

> **RouterD(config)#** *ntp master 10*
> **RouterD(config)#** *ntp authentication-key 1 md5 MYPASSWORD*

RouterD is the NTP server for RouterC. To secure this association, the following must be configured on RouterC:

> **RouterC(config)#** *ntp authenticate*
> **RouterC(config)#** *ntp authentication-key 1 md5 MYPASSWORD*
> **RouterC(config)#** *ntp trusted-key 1*
> **RouterC(config)#** *ntp server 50.1.1.1 key 1*

Usually in client/server environments, the server authenticates the client. However, NTP *clients* authenticate the NTP *server.*

The *authentication-key* must be the same on both NTP devices.

Misc. NTP Commands

You may have noticed that, in client/server NTP environments, *no* configuration is necessary on the NTP server to "service" clients. Thus, any device can point to a Cisco router as its NTP server.

If this is undesirable, NTP can be disabled on a specific interface:

> **Router(config)#** *interface fa0/1*
> **Router(config-if)#** *ntp disable*

To restrict the number of client "associations" to an NTP server:

> **Router(config)#** *ntp max-associations 10*

To provide stability to the NTP network, the *source* interface of NTP packets can be specified. It is usually preferable to use a loopback interface as a source, as loopbacks should never be in a *down* state:

> **Router(config)#** *ntp source loopback0*

Verifying NTP

To view the current date and time:

> **Router#** *show clock*
>
> 04:30:03.871 CDT Mon Jun 27 2005

To view the status of NTP synchronization:

> **Router#** *show ntp status*
>
> Clock is synchronized, stratum 5, reference is 10.1.1.1
> nominal freq is 250.0000 Hz, actual freq is 249.9941 Hz, precision is 2**18
> reference time is BF454660.7CCA9683 (04:30:03.871 CDT Mon Jun 27 2005)
> clock offset is 3.1328 msec, root delay is 141.22 msec
> root dispersion is 29.39 msec, peer dispersion is 1.22 msec

To view the NTP associations:

Router# *show ntp associations*

```
address          ref clock      st  when  poll reach  delay  offset  disp
*~10.1.1.1       133.4.22.22     3     3    64   368    66.2    3.44   0.3

 * master (synced), # master (unsynced), + selected, – candidate, ~
configured
```