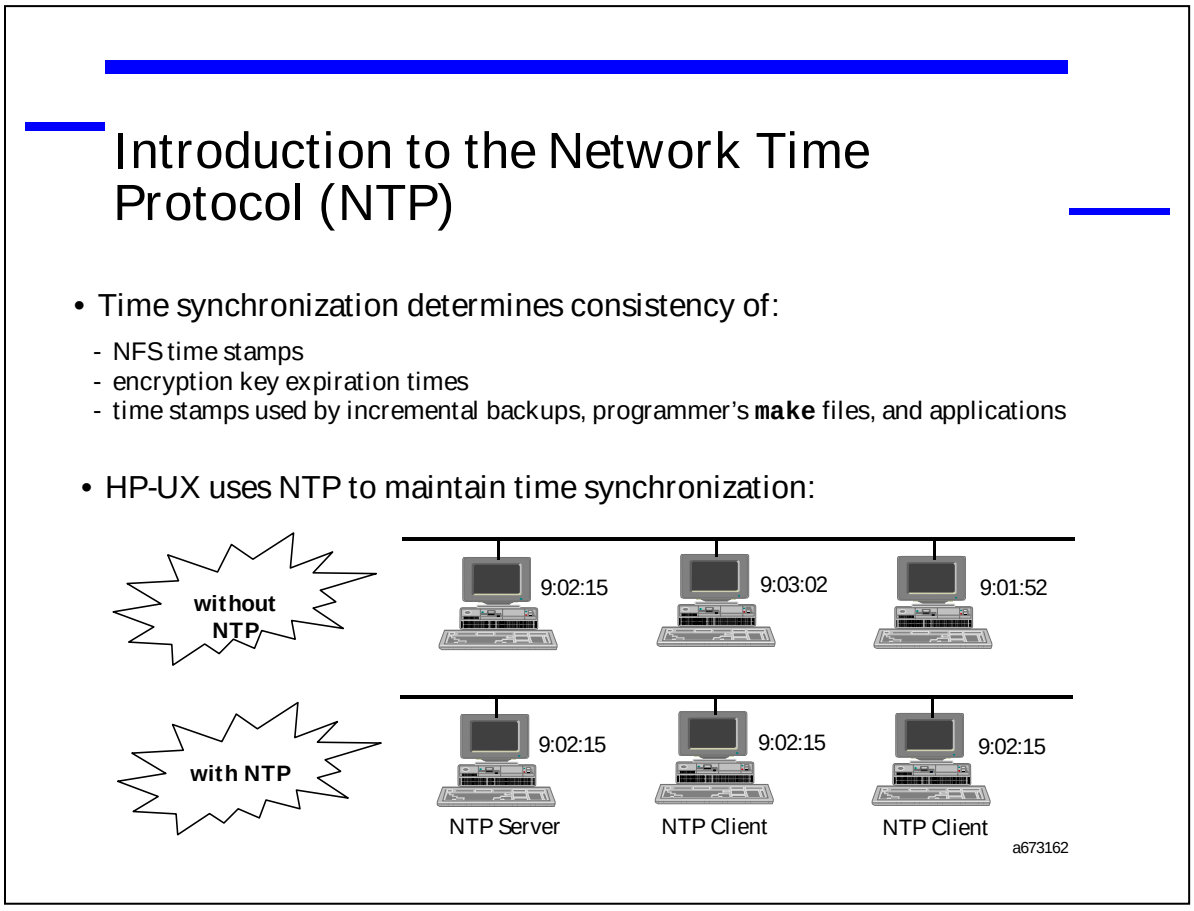
Module 15 — Introduction to the Network Time Protocol (NTP)

Objectives

Upon completion of this module, you will be able to do the following:

- List three reasons for implementing network time synchronization.
- Describe the NTP **stratum level** concept.
- Define the following terms:
  - NTP server
  - NTP peer
  - NTP broadcast client
  - NTP polling client
- Configure an NTP server.
- Configure an NTP broadcast client.
- Configure an NTP direct-poll client.
- Monitor NTP using the **`ntpq`** command.

15–1. SLIDE: Introduction to Network Time Protocol (NTP)



Student Notes

In a networking environment, synchronize the clocks of all members for the following purposes:

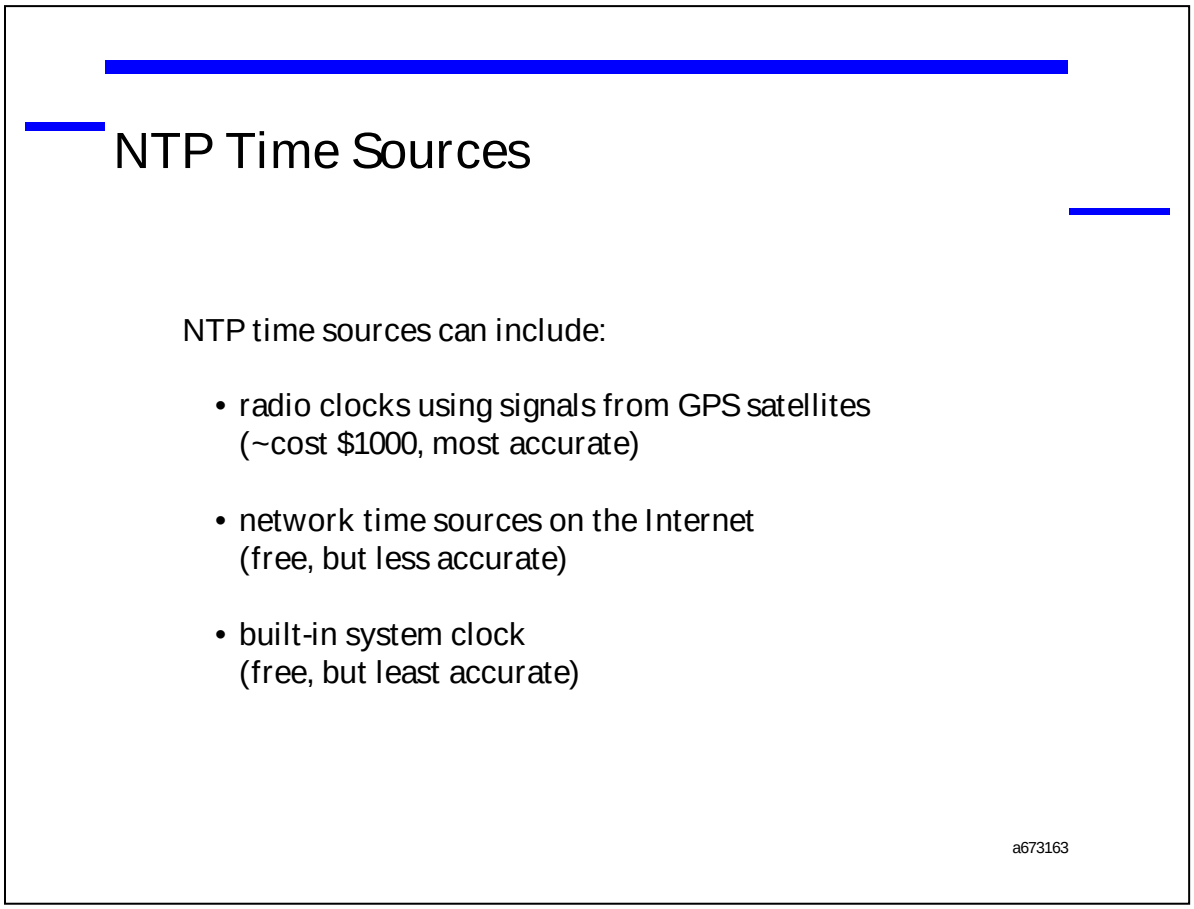
- To perform incremental backups properly between the client systems and the backup server.
- To secure RPCs which utilize Kerberos tickets between two systems. A ticket is valid for only a certain period of time in order to prevent reuse. If the clocks are not synchronized, a ticket can expire prior to being used.
- For applications that need to know when a user last logged in, or when a file was last modified.

In certain network computing environments a system time difference of one minute or less is too much. In large environments, you cannot meet this requirement by managing the system times manually with the **`date`** command.

With HP-UX, a network time service is bundled with the operating system. It is called Network Time Protocol (NTP) and is used to synchronize system times in the network. The **`xntpd`** daemon is used to implement this new feature.

NTP is configurable through the command line or through SAM.

15–2. SLIDE: NTP Time Sources



Student Notes

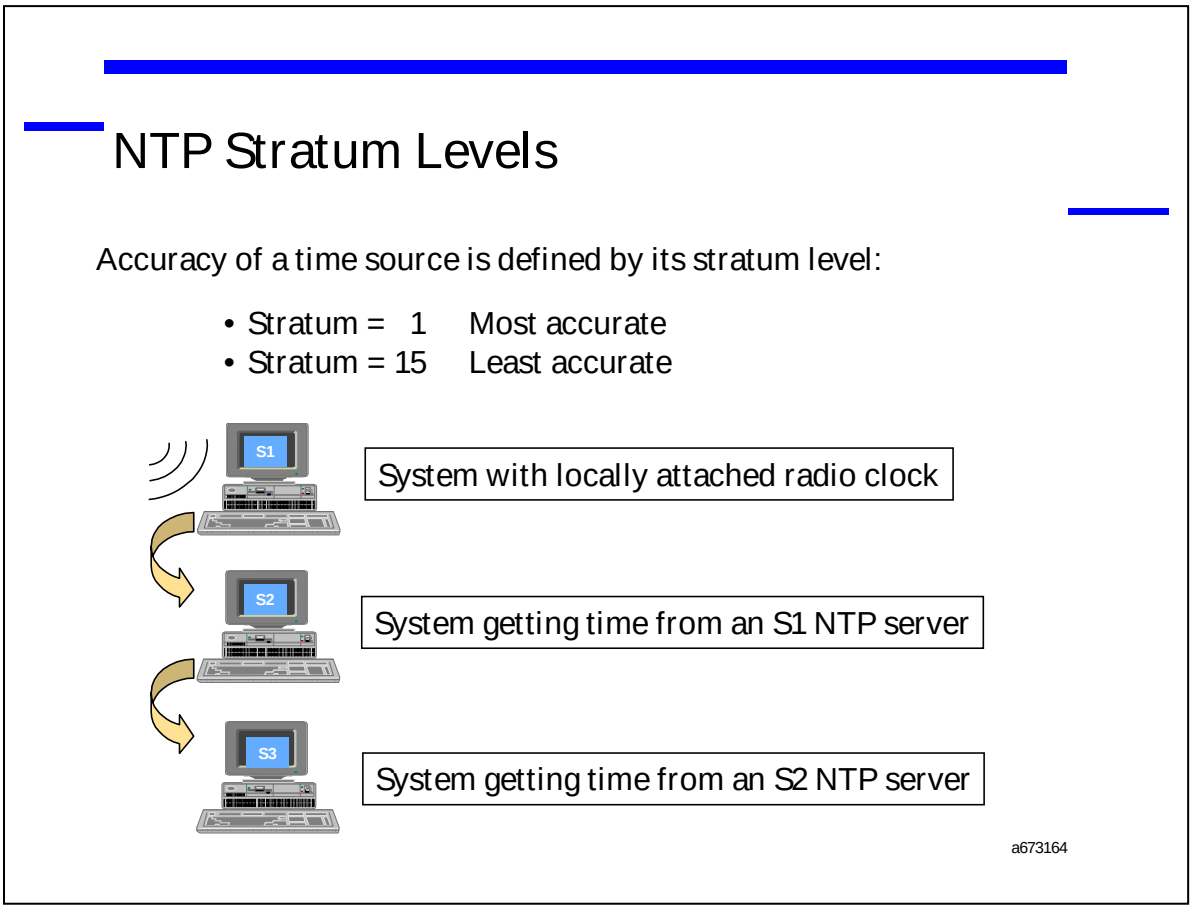
NTP can be used to synchronize system clocks using a variety of time sources:

- A radio clock can be attached to the serial port of an HP-UX system. A radio clock determines the current time using signals from GPS (Global Positioning System) satellites. Radio clocks are one of the most accurate time sources, but the cost ranges from hundreds to a thousand dollars. A list of radio clock suppliers is available at **`http://www.eesic.udel.edu/~ntp`**.

- A NTP time server on the network can be used to synchronize a system's clock, if the cost of a radio clock is too expensive. A list of stratum 1 NTP time servers on the public internet is available from **`http://www.eesic.udel.edu/~ntp`**.

- The internal clock of any system on a local network can be used if time synchronization is only needed among the nodes on the local network. One node would have to be selected as the authoritative time source, and then the other nodes would simply synchronize their times with the authoritative node. Using this method, all nodes would agree on a common system time, but would not necessarily be synchronized with other nodes beyond the local network. If a system is on a local area network and can't afford a radio clock, this is the best option.

15–3. SLIDE: NTP Stratum Levels

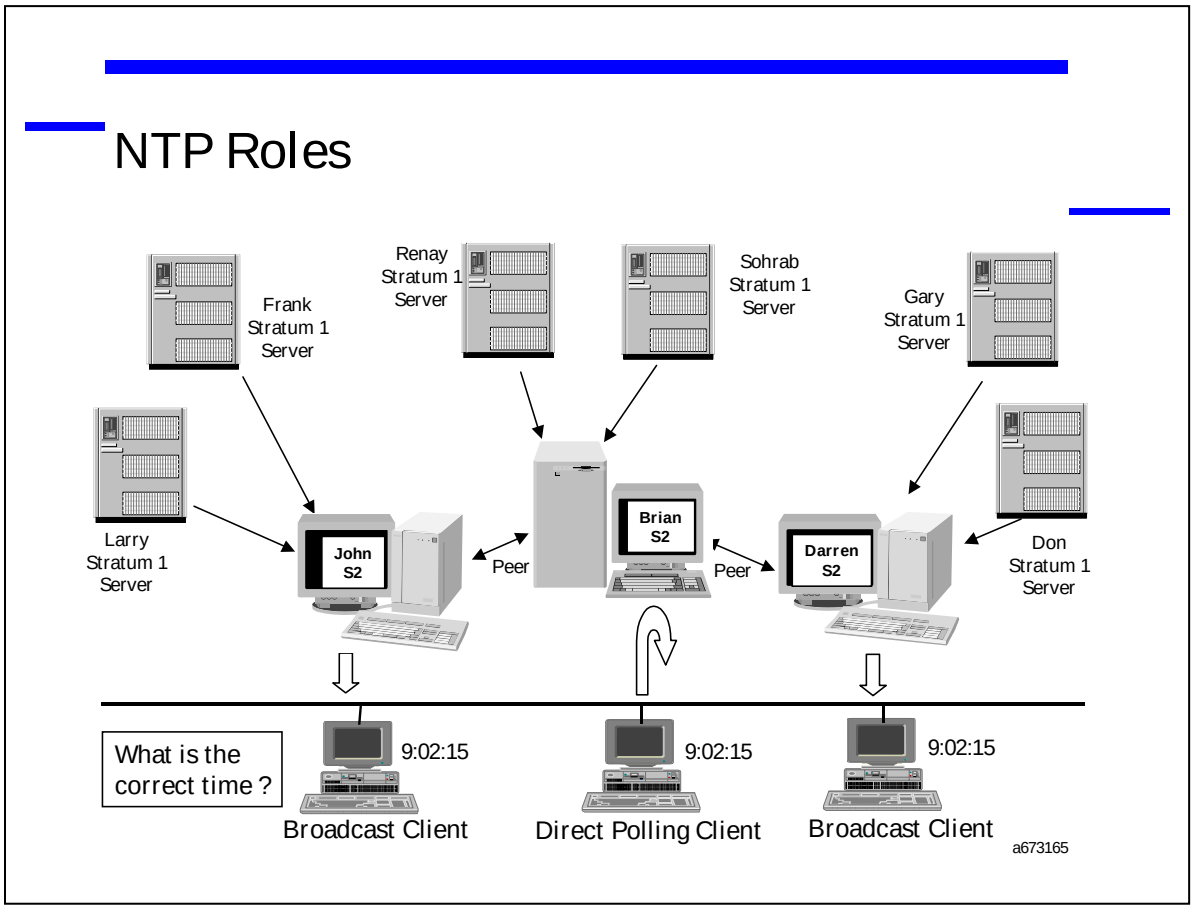


Student Notes

In a large network, several hierarchically organized timeservers can be used to synchronize the clocks of all systems on the network. The highest level time servers (stratum 1) receives time signals directly from an external time source (such as a radio clock). The time servers on the next levels estimate the correct time by consulting one or more time servers at equal or higher stratum levels (with 1 being the highest stratum level).

Each time server continually updates its notion of the current time by examining timestamps from remote servers, compensating for unpredictable network delays. For the lowest stratum levels in the hierarchy (the highest stratum numbers), the systems are pure clients, which only receive time signals from time servers.

15–4. SLIDE: NTP Roles



Student Notes

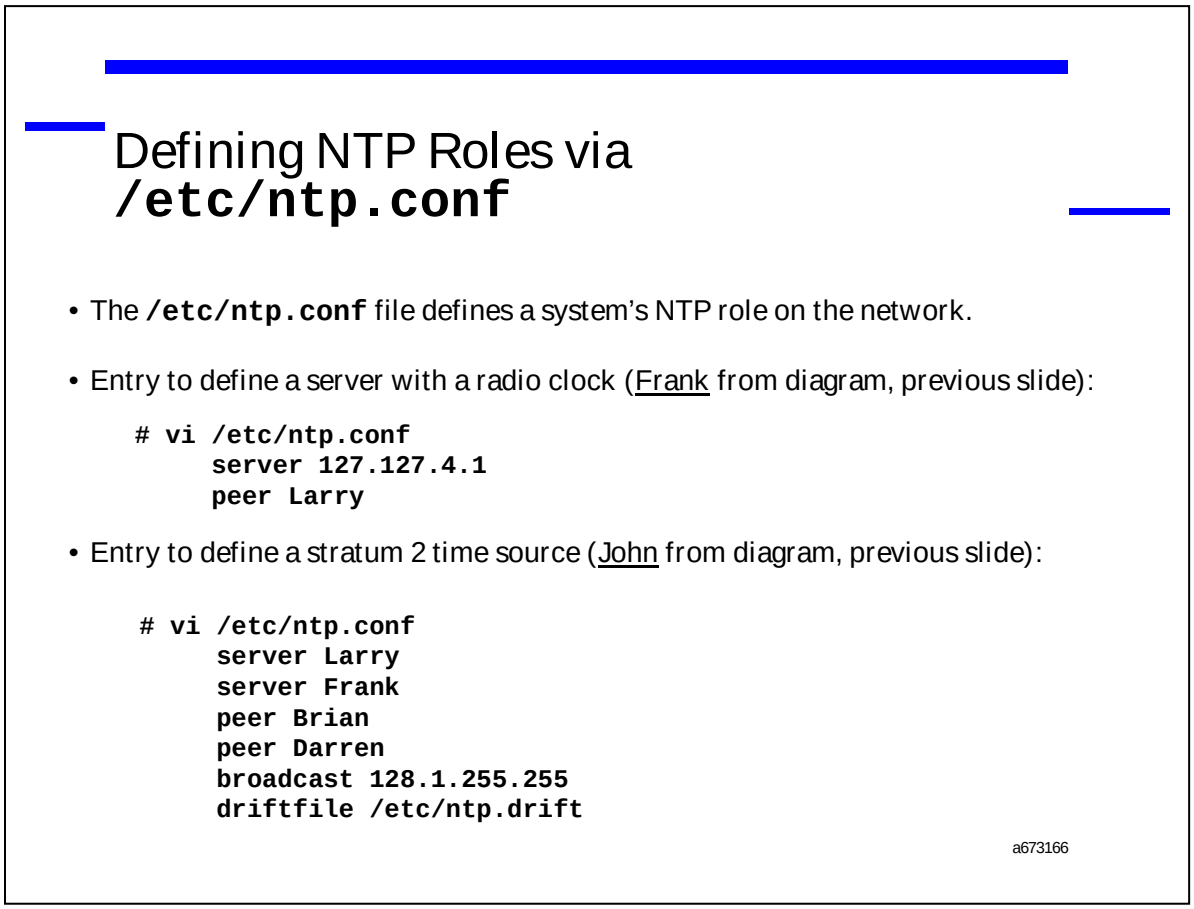
When implementing NTP on a network, there are four possible roles a system could play:

server	An NTP server provides time to other systems.
peer	An NTP peer obtains its time from one or more NTP servers usually of equal time stratum with itself. NTP uses a special algorithm to reconcile any time differences from the different servers and its own local time into a single time. This time is considered to be the most accurate and is based on the stratum levels of the other servers as well as other conditions.
direct polling client	A direct polling client receives its time directly from a specific time server by sending a time poll request to the time server.
broadcast client	A broadcast client listens for time requests being broadcast by NTP time servers.

In the example on the slide, system John is served by machines Larry and Frank (both stratum 1 NTP servers). NTP on John resolves any time differences (at most, milliseconds)

into a single time on John. If John loses connection to Larry, it gets its time solely from Frank. If John loses connection to both Larry and Frank, it then queries its peers (Brian and Darren) to determine the time.

15–5. SLIDE: Defining Roles via `/etc/ntp.conf`



Student Notes

The **`/etc/ntp.conf`** file is used to define a system's relationship (relative to NTP) with other systems on the network. This relationship is defined at boot time when the system starts up.

Configuration for a Stratum–1 Server with a Radio Clock

To configure a stratum1 server, add the following lines to the **`/etc/ntp.conf`** file (example for server Frank):

```
server 127.127.4.1
peer Larry
```

Note regarding the above entry:

- The IP address is a pseudo IP address that identifies a radio clock time source.
- The three octet combination, 127.127.4 is used within NTP to define the time source as a radio clock.

- The fourth octet is a value 1 through 4 which identifies the device file for the clock. Device files for radio clocks are of the form **`/dev/wwvb1`**, where the last digit of the device file name is the same as the fourth octet of the IP address.

- Each radio clock server should peer with several other stratum-1 servers in case the local radio clock becomes unavailable. The entry **`peer Larry`** is such an example.

Configuration for a Stratum–2 Server

Below is an example of NTP configured for stratum-2 server John from the previous diagram. Note that John also peers with systems Brian and Darren (in case the stratum-1 servers become unavailable), and John acts as a broadcast time server for other local clients on the same network.

```
server Larry
server Frank
peer Brian
peer Darren
broadcast 128.1.255.255
driftfile /etc/ntp.drift
```

Notes regarding the above entry:

- The server entries define lower stratum servers to poll.

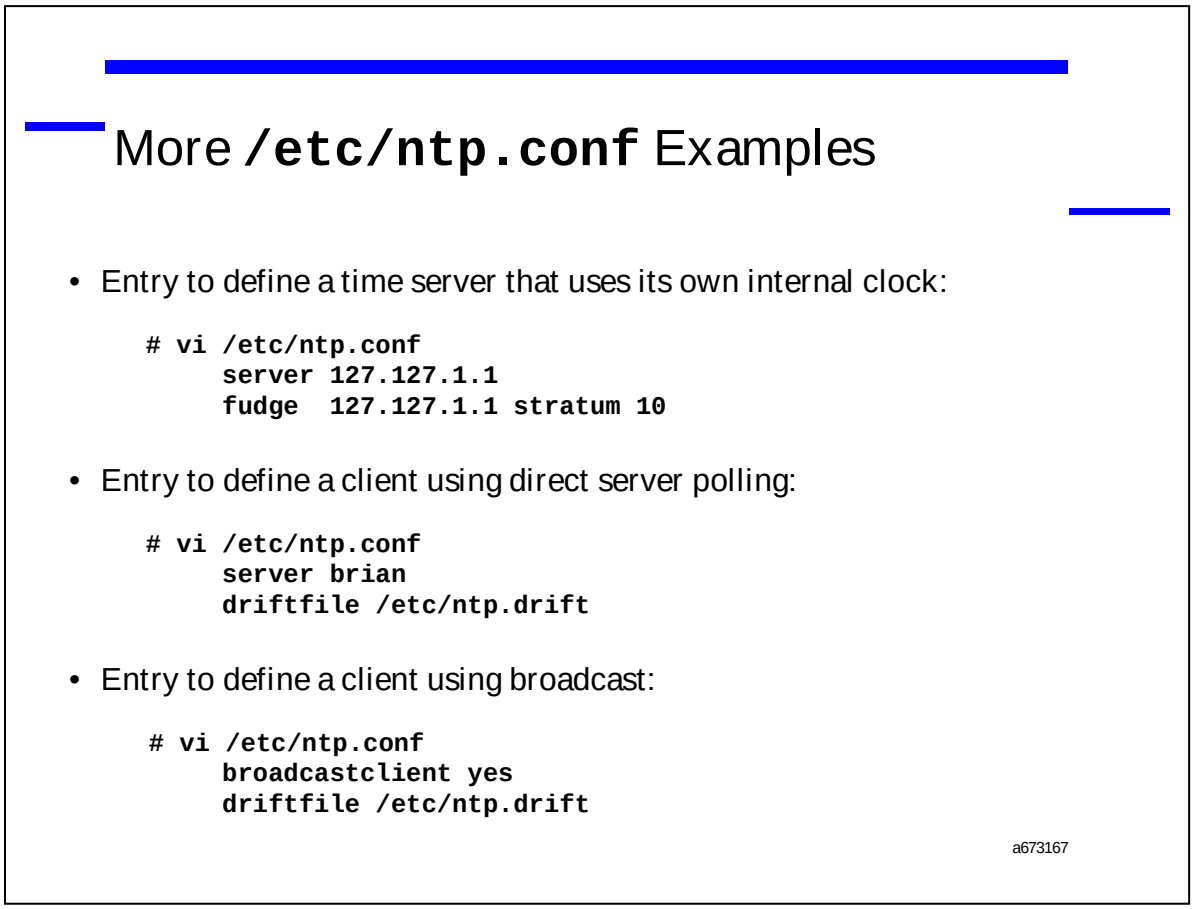
- The peer entries define other servers to poll at the same stratum levels.

- The broadcast entry defines the broadcast address which NTP should use to broadcast the time on the network.

- The driftfile entry specifies the name of a file to use to track long-term drift of the local clock.

- Over time, NTP will use the driftfile to compensate for drift, and will poll NTP servers less frequently.

15–6. SLIDE: More `/etc/ntp.conf` Examples



Student Notes

Below are some additional NTP configuration examples:

Configuration for a Local NTP Server Using its Internal Clock

To configure an NTP server to use its own system clock as an authoritative time source, add the following lines to the server's **`/etc/ntp.conf`** file:

```
server 127.127.1.1
fudge 127.127.1.1 stratum 10
```

Notes regarding the above entry:

- The IP address is a psuedo IP address that identifies the local system as a time source.
- The specific IP address of 127.127.1.1 tells NTP to use the internal clock as the time source.
- The fudge entry defines a stratum level to be assigned to this clock. If no fudge entry is used, the node will be defined as a stratum-1 server. It is a good idea to reassign the stratum level to 10 so clients polling this server will know it is not a reliable time source.

- This method of time synchronization should only be used on localized networks with no access to an external time source.

Configuration for a Client using Direct Server Polling

To configure a client to poll a specific NTP server, add the following line to the client's **`/etc/ntp.conf`** file:

```
server brian
driftfile /etc/ntp.drift
```

Notes regarding the above entry:

- The client will poll the server brian to validate its time.
- The default polling interval starts at 64 seconds and adjusts from there.
- The driftfile is used to track differences between the client's time and the server's time. As the driftfile stabilizes it will be used more frequently to adjust the client's time, and the server will be polled less frequently.

Configuration for a Client using Broadcast Polling

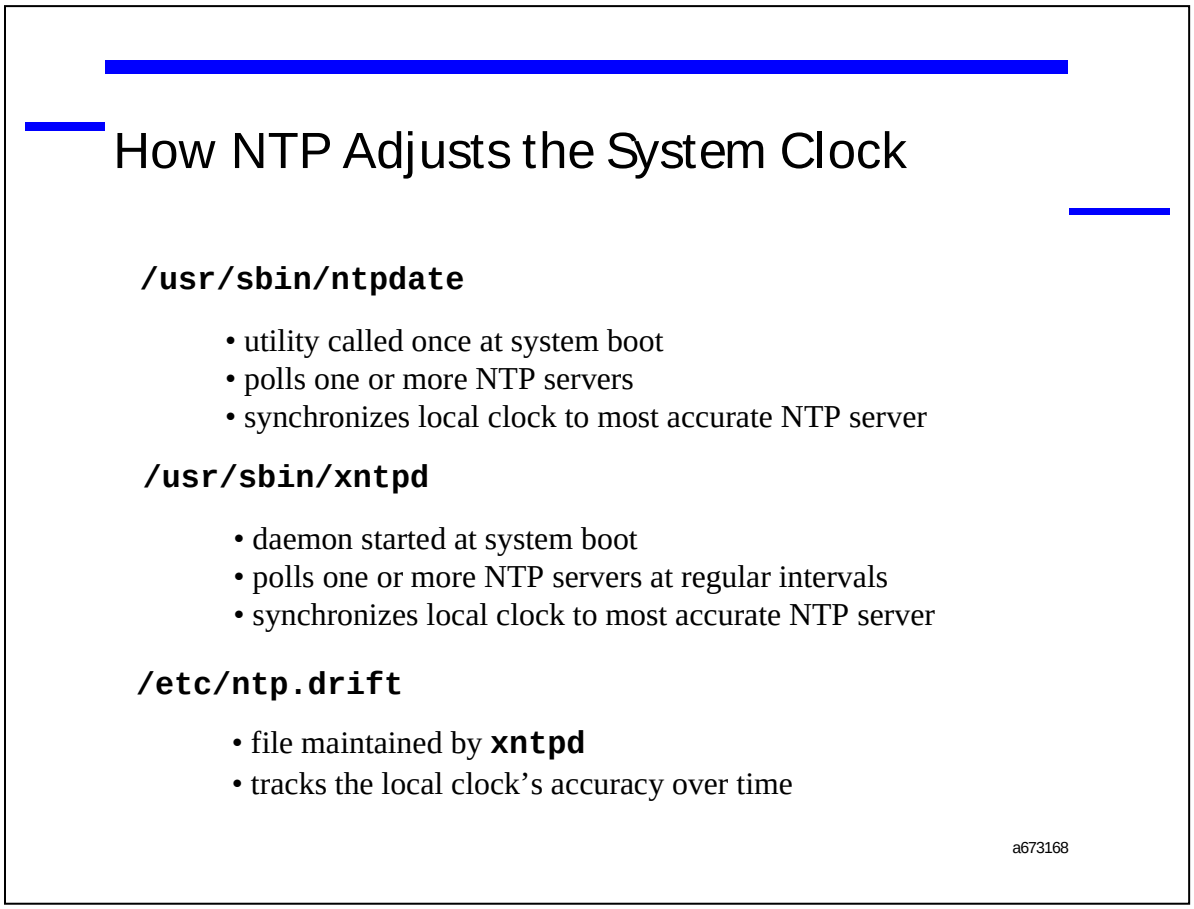
To configure a client to listens for time broadcasts, add the following line to the client's **`/etc/ntp.conf`** file:

```
broadcastclient yes
driftfile /etc/ntp.drift
```

Notes regarding the above entry:

- The client will passively listen for NTP broadcasts and adjust his clock appropriately.
- This method is recommended over direct server polling for a large network with lots of nodes needing to be time-synchronized. This method is recommended because it significantly reduces the traffic load related to NTP.
- Clients must be on the same subnet as the NTP broadcast server.

15–7. SLIDE: How NTP Adjusts the System Clock



Student Notes

There are three different time mechanisms which NTP uses to keep clocks in sync with other nodes on the network.

The `ntpdate` Command

The **`ntpdate`** command is used to immediately set a client's clock to be in sync with the time servers on a network. This is the quickest way to get a client's clock in sync with the NTP server's time.

This command is used most often at boot time to ensure the client's clock is in sync with the time on the network.

This command is normally used with one NTP server specified as an argument, and the client's clock is set to match the time of that NTP server. However, if multiple NTP servers are specified with this command, then the client uses the NTP server with the lowest stratum. If the stratum levels are equal, then the client uses an average of the lowest stratum NTP servers.

The `xntpd` Daemon

The **`xntpd`** daemon runs continuously in the background and constantly verifies the client clock is in sync with the NTP servers on the network. The **`xntpd`** daemon is normally started up at boot time.

Because client's clocks can drift from the actual NTP clocks over time, the **`xntpd`** daemon periodically performs a network query of the NTP server's clock, compare the time with the client's clock, corrects the client's time as necessary, and records the time difference in a file, **`/etc/ntp.conf`**.

The default time interval for **`xntpd`** time checking is once every 64 seconds. At the first check, the delta between the client's clock and NTP server's clock is recorded and then depending on the time difference, the next time check is scheduled. If the time difference is high, then the next time check will occur soon. If the time difference is small, then the next time check will be scheduled further out.

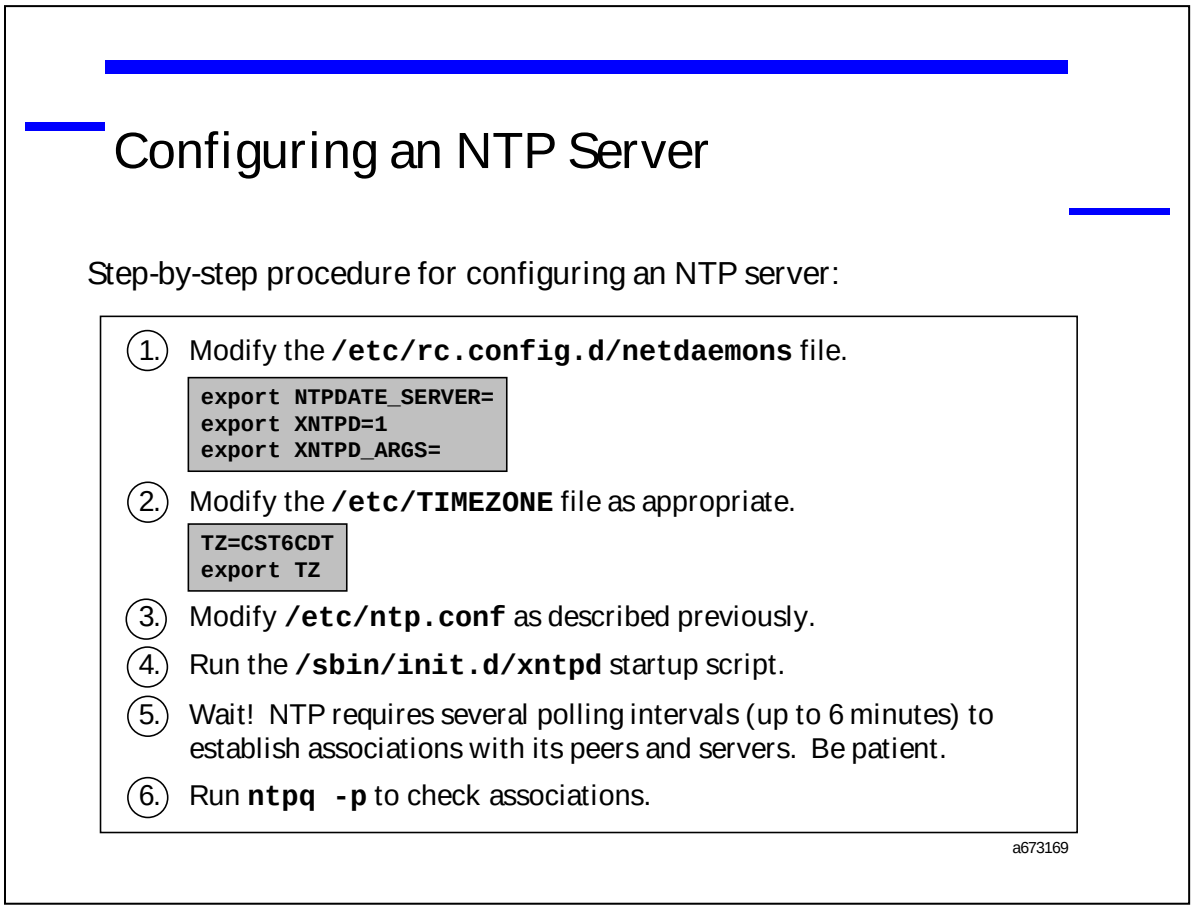
The `/etc/ntp.conf` File

The file, **`/etc/ntp.conf`**, is used to record the deltas (or drifts) between the client's clock and the NTP server clock at each time check interval.

It is expected that over time, the difference between the client and NTP server clocks will be consistent from time interval to time interval. Once the **`xntpd`** daemon determines the anticipated drift (or difference) between the client and the NTP server's clocks (for a specific time interval), the **`xntpd`** daemon will begin to use the expected drift, which is recorded in the file, **`/etc/ntp.conf`**, rather than query the NTP time server over the network. By doing this, the client reduces the traffic on the network.

The **`xntpd`** daemon slowly switches over to referencing the drift file, rather than querying the NTP servers. As the drift value stabilizes and remains the same, the querying of the NTP servers will become less and less frequent, until the minimal amount of NTP querying is reached.

15–8. SLIDE: Configuring an NTP Server



Student Notes

The steps to configure a system to be an NTP Server are:

1. Edit the **/etc/rc.config.d/netdaemons** file to configure the **xntpd** daemon to startup every time the system boots. Set the XNTPD variable to equal 1

   ```
   export NTPDATE_SERVER=
   export XNTPD=1
   export XNTPD_ARGS=
   ```

   For NTP servers, do not set the NTPDATE_SERVER variable (leave it blank). This variable is for NTP clients.

2. Edit the **/etc/TIMEZONE** file and specify the correct time zone for the system. Set the TZ variable to equal the time zone for the system. See the **/usr/lib/tztab** file for a list of all the available time zones.

   ```
   TZ=CST6CDT
   export TZ
   ```

3. Edit the **/etc/ntp.conf** file and define the NTP server (as described earlier in this module). Below is an example for defining an NTP server to use its local clock:

```
server 127.127.1.1
fudge 127.127.1.1 stratum 10
```

4. Start the **xntpd** daemon manually by executing the following command:

   ```
   sbin/init.d/xntpd start
   ```

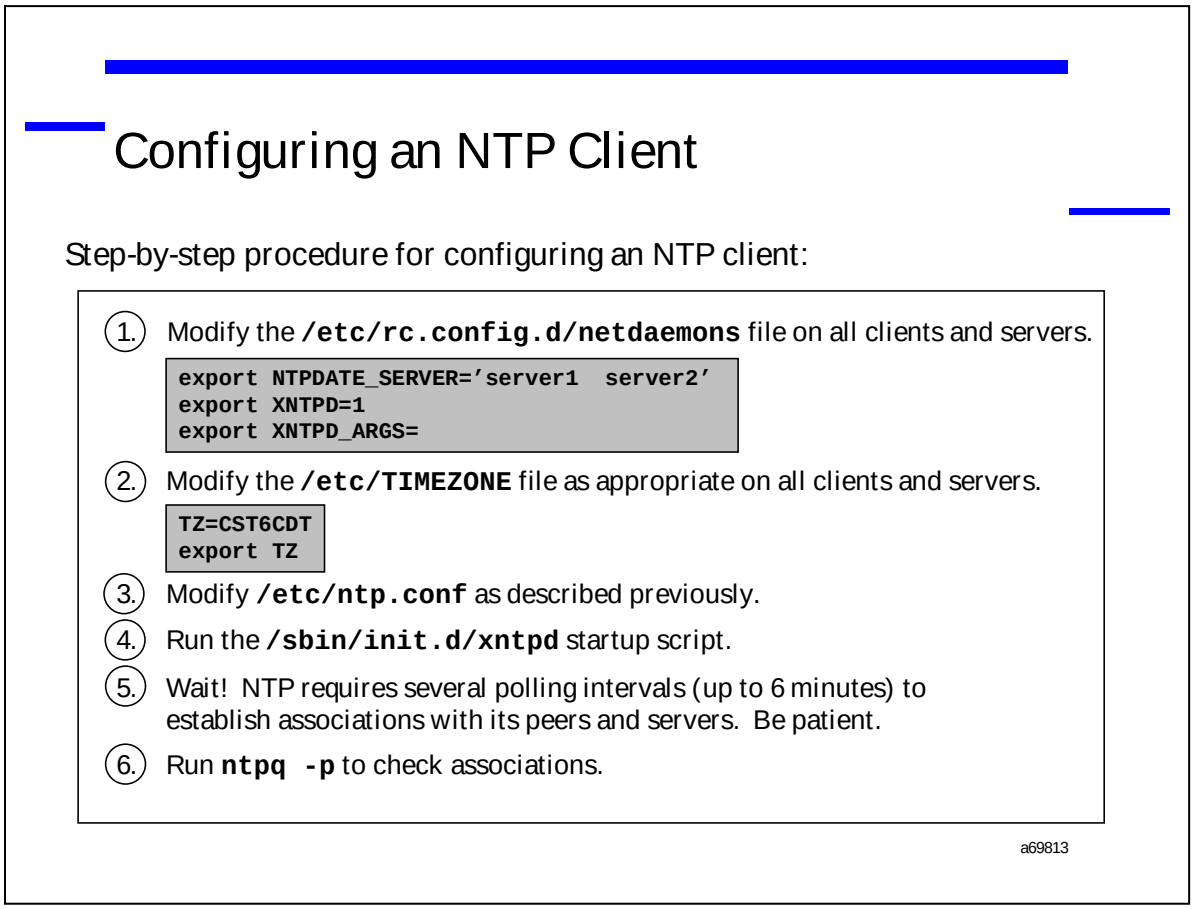
5. Wait. It could take up to 6 minutes for the **xntpd** daemon to start.

6. Verify the NTP server configuration (and its association with peer NTP servers) by executing the following command:

   ```
   ntpq -p
   ```

   More information on the **ntpq** command is contained in the upcoming slides.

15–9. SLIDE: Configuring an NTP Client



Student Notes

The procedure for configuring an NTP client is virtually identical to that of configuring an NTP server — only the contents of the configuration files change.

The complete, step-by-step procedure for configuring an NTP client is:

1. Edit the **/etc/rc.config.d/netdaemons** file to configure the **xntpd** daemon to startup every time the system boots. Set the XNTPD variable to 1, and specify which NTP servers to query when the **ntpdate** command is used:

   ```
   export NTPDATE_SERVER='NTP_server1 NTP_server2'
   export XNTPD=1
   export XNTPD_ARGS=
   ```

2. Edit the **/etc/TIMEZONE** file and specify the correct time zone for the client system. See the **/usr/lib/tztab** file for a list of all the available time zones.

   ```
   TZ=CST6CDT
   export TZ
   ```

3. Edit the **/etc/ntp.conf** file and define the NTP client (as described earlier in this module). Below is an example for defining an NTP broadcast client:

```
broadcastclient yes
driftfile /etc/ntp.drift
```

4. Start the **xntpd** daemon manually by executing the following command:

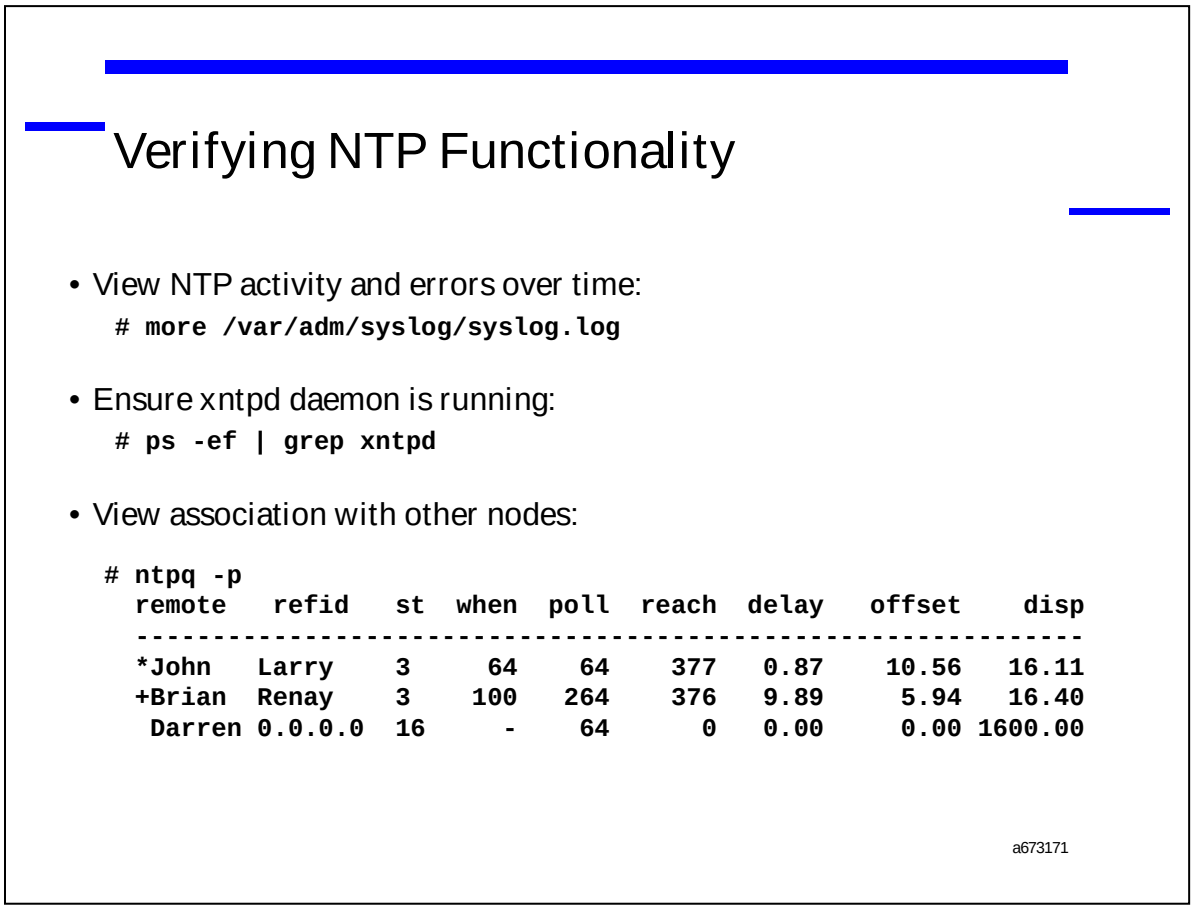
   ```
   /sbin/init.d/xntpd start
   ```

5. Wait for the **xntpd** daemon to start. It could take up to 6 minutes for the daemon to establish an association with the corresponding NTP servers and peers.

6. Verify association with other NTP server and peers were correctly established. Execute the command:

   ```
   ntpq -p
   ```

15–10. SLIDE: Verifying NTP Functionality



Student Notes

When troubleshooting NTP, the three recommendations are:

- Check the syslog file.

  When the **xntpd** daemon starts up, it logs a number of entries to the **/var/adm/syslog/syslog.log** file, including:

  - Timestamps of when the **xntpd** was started and stopped.
  - Associations formed with other nodes running NTP.
  - Errors found in the **/etc/ntp.conf** file.

- Verify that the **xntpd** daemon is running.

  The **xntpd** daemon is needed to ensure time accuracy on the client. If this daemon is not present, then no time synchronization with other NTP systems is being performed.

- View status of NTP server nodes and their association with the client.

  The command

`ntpq -p`

queries for NTP servers on the network, and displays the association which the client has with each NTP server. For example

```
# ntpq -p
remote   refid      st     when  poll  reach  delay   offset     disp
------------------------------------------------------------------------
*John    Larry      3        64    64    377   0.87    10.56    16.11
+Brian   Renay      3       100   264    376   9.89     5.94    16.40
 Darren  0.0.0.0    15        -    64      0   0.00     0.00  1600.00
```

*****	Indicates the NTP server responded and was chosen as most accurate
+	Indicates the NTP server responded to the query request
blank	Indicates the NTP server did not respond to the query request
remote	The name of the NTP server which responded to the query request
refid	The name of the higher level stratum server used by the NTP server
st	The stratum level of the responding NTP server
when	The number of seconds since the last successful poll
poll	The number of seconds being used for the current polling interval
Others	Other statistics used to determine accuracy and disparity

15–11. LAB: Introduction to NTP

Directions

Your instructor will assign you to work with a team of your classmates to configure an NTP server, and one or more NTP clients. Record the hostnames and chosen roles of your teammates' machines below.

```
NTP server: ____________

NTP client: _____________

NTP client: _____________
```

Record the commands you use to complete the steps below, and answer all questions.

Part 1: Configuring an NTP Server

The steps below should only be configured on the host you have chosen to be the NTP server. Do not start configuring the NTP clients until the server configuration is complete.

Since you probably don't have access to a radio clock in the classroom, use the NTP server's internal system clock as the authoritative time source for your team.

1. Set the local clock forward 2 minutes so the clients can actually see a clock "step" after enabling NTP.

   ```
   date MMDDhhmm
   ```

2. Add a server line to the end of the **`/etc/ntp.conf`** file defining the local clock as the only time source. Since the internal system clock isn't likely to be accurate, set the stratum level of this time source to 10.

   ```
   # vi /etc/ntp.conf
         server 127.127.1.1
         fudge 127.127.1.1 stratum 10
   ```

3. Modify the **`/etc/rc.config.d/netdaemons`** file to enable XNTPD on the server. Don't specify an NTP date server.

```
# vi /etc/rc.config.d/netdaemons
      NTPDATE_SERVER=
      XNTPD=1
      XNTPD_ARGS=
```

4. Run the NTP startup script to start the **xntpd** daemon.

   ```
   # /sbin/init.d/xntpd start
   ```

5. After **xntpd** starts, it takes 5 poll cycles (320 seconds) to establish the appropriate peer and server relationships. Wait 5 minutes before proceeding on to the next question.

6. Is the **xntpd** daemon running? Are there any NTP errors in the syslog?

   ```
   # ps -ef | grep xntpd
   ```

   ```
   # tail /var/adm/syslog/syslog.log
   ```

   If all is well, the daemon should be running, and there shouldn't be any XNTPD "ERROR"s in the syslog.

7. Does **ntpq -p** suggest that the correct association has been formed? What stratum level did NTP assign to your local clock?

   ```
   # ntpq -p
   ```

   There should be one line in the **ntpq -p** output showing that the local clock is being used as a stratum 10 time source.

Part 2: Configuring an NTP Client

Don't start this procedure until you confirm that your NTP server is fully functional. The steps below should only be performed on the host(s) you have chosen as NTP clients.

1. Add appropriate server and driftfile lines to your **`/etc/ntp.conf`** file to poll the NTP server created in the previous portion of the exercise.

   ```
   vi /etc/ntp.conf
   server 128.1.1.1         # assume 128.1.1.1 is the NTP srvr IP
   driftfile /etc/ntp.drift
   ```

   You may use the server's hostname rather than the IP if you wish.

2. Modify the **`/etc/rc.config.d/netdaemons`** file to enable **`xntpd`**. Also define your NTP server to be the NTPDATE_SERVER.

   ```
   # vi /etc/rc.config.d/netdaemons
         NTPDATE_SERVER=128.1.1.1   # Assume 128.1.1.1 is the NTP srvr IP
         XNTPD=1
         XNTPD_ARGS=
   ```

   Here again, you may use the server's hostname in place of the IP if you wish.

3. Run the NTP startup script on the client to start the NTP daemon. Note the output as **`ntpdate`** steps the system clock.

   ```
   # /sbin/init.d/xntpd start
   ```

4. Check to ensure that your client formed the proper association by running **`ntpq -p`**.

   ```
   # ntpq -p
   ```

5. Compare the time on your client against the time on the NTP server. Do they appear to be synchronized at this point?

   ```
   Execute the date command on both machines.
   They should agree.
   ```