Clock Synchronization

It is important that the system clocks in the Linux workstations of the RFPK Software Team be closely synchronized with the clock in the Linux server, whitechuck, because the CVS application on that machine as well as the backup process depends on the comparison of timestamps on files residing on different machines. This tutorial explains how to configure your workstation so that it stays synchronized within a fraction of a second with that on whitechuck.

Table of Contents

Network Time Protocol

The Network Time Protocol (NTP) allows computers connected to the Internet to keep their clocks synchronized. Certain host computers equipped with accurate clocks make the NTP service available to other, less well endowed, hosts. A machine with an accurate clock which is offering an NTP service is known as a *level one* NTP server. Other machines, which may not be equipped with accurate clocks but which nevertheless have accurate time due to the fact that they are synchronized to one or more *level one* NTP servers, can offer an NTP service over the Internet as *level two* NTP servers, and so on.

The host bigben.cac.washington.edu is a *level one* NTP server equipped with a clock that gets its time from GPS satellites. The RFPK Linux server, whitechuck, is synchronized to bigben and several other *level one* NTP servers. This document describes the way to configure your RedHat Linux 8.0 workstation to synchronize to whitechuck.

There are two reasons for synchronizing our workstations to whitechuck rather than to bigben:

- Reduce the load on bigben.
- Synchronize to the machine that we really want to be in sync with. Even if whitechuck should go out of sync with the *level one* servers because of Internet problems, for example, our CVS and backup processes would still work well because they would be comparing timestamps on files on our workstations with those on whitechuck.

NTP Configuration

In this section, we will edit several configuration files. To do this, you will need root privilege.

We will start by putting the file `/etc/ntp.conf` under local source control:

```
su -
cd /etc
ci -l /etc/ntp.conf
```

With your favorite editor, set the default behavior to *nomodify* rather than *ignore*, which completely disabled the NTP service.

```
restrict default ignore

 should be changed to

restrict default nomodify
```

Next specify that whitechuck is our trusted time server by making these changes:

```
# restrict mytrustedtimeserverip mask 255.255.255.255 nomodify notap noquery
# server mytrustedtimeserverip

 should be changed to

restrict whitechuck mask 255.255.255.255 nomodify notrap noquery
server whitechuck
```

Finally set up a dummy time server in case the Internet is down:

```
server time.nist.gov

 should be replaced by

server 127.127.1.0
```

After making the above changes, use the **rcsdiff** command to check your work:

```
rcsdiff ntp.conf
```

Finally, we need to make a small change to the `/etc/ntp/step-tickers` file, which is a list of time servers queried by your system when it boots.

Still as root, place `/etc/step-tickers` under source control:

```
cd /etc/ntp
ci -l step-tickers
```

Now we will concatenate the name whitechuck to the list of time servers to consulted at start up:

```
echo whitechuck >> step-tickers
```

Daemon Setup

We need to start the ntp daemon, if it is not already running, and make sure that it is started automatically whenever the system reboots. As root, input the following commands:

```
/etc/rc.d/init.d/ntpd restart
chkconfig --level 2345 ntpd on
```

You should now check that your ntp daemon is working. Issue this command:

```
ntpq -p
```

You should see output that looks something like this:

```
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 LOCAL(0)        LOCAL(0)        10 l   63   64  377    0.000    0.000   0.008
*whitechuck.rfpk bigben.cac.wash  2 u  192  256  377    0.684    0.075   0.024
```

I believe that this shows that there are two time servers, the dummy local server to be used in case the network is down, and whitechuck. It shows that whitechuck gets its time from bigben, whitechuck is a level 2 server, it is remote, it was last queried 192 seconds ago, it is queried every 256 seconds, a measure of its estimated "reachability" is 377, there is a delay of 0.684 milli-seconds between the time it is queried and the time the response arrives, an offset of 0.075 milli-seconds was applied in order to synchronize the clocks and an estimate of the random error is 0.024 milli-seconds.

If you do not get comparable numbers (for example, the jitter is 0.000) there is something wrong.