

Synchronize your network with NTP

NTP (*Network Time Protocol*) allows clock synchronization between computer systems. The following HOWTO describes:

- configuring an NTP server on Slackware Linux;
- synchronizing client PCs with your local NTP server.

Introduction

When several users manipulate shared data on different client PCs on a network, it's important that these machines are all synchronized. This is especially true if you share files over NFS, or if you use NIS for centralized authentication. You'll get all sorts of weird errors if your clocks are out of sync. Unfortunately, the clients' onboard clocks aren't sufficiently precise.

That's where NTP (*Network Time Protocol*) comes in handy. It allows networked machines to adjust their clocks so as to be perfectly synchronized. A series of public time servers on the Internet allow the reception of the exact time. From this point, we can use NTP in several ways.

- The `ntpdate` command makes an initial correction of the BIOS clock.
- This one-time-adjustment isn't sufficient for a server that is supposed to be up 24/7, since its clock will drift away gradually from the exact time. In that case, we have to configure the `ntpd` daemon (shipping with the `ntp` package). This daemon contacts public time servers at regular intervals and proceeds with incremental corrections of the local clock.
- The `ntpd` daemon can in its turn be configured as a time server for the local client machines.

It's considered good practice to use `ntpdate` for the initial adjustment and `ntpd` for regular time synchronization.

Firewall considerations

The NTP services uses UDP port 123. Open this port if you want to allow remote machines to connect to your NTP server.

Synchronize a LAN server or a public root server with an NTP server on the Internet

Create an empty log file:

```
# touch /var/log/ntp.log
```

Visit <http://www.pool.ntp.org> and choose a list of servers according to your country.

Configure the NTP service by editing `/etc/ntp.conf`. You might backup the existing `ntp.conf` file

and start from scratch.

In the example below, the list of four servers is chosen for my company's location (France):

```
# /etc/ntp.conf

driftfile /etc/ntp/drift
logfile /var/log/ntp.log

server 0.fr.pool.ntp.org
server 1.fr.pool.ntp.org
server 2.fr.pool.ntp.org
server 3.fr.pool.ntp.org

server 127.127.1.0
fudge 127.127.1.0 stratum 10

restrict default nomodify nopeer notrap
restrict 127.0.0.1 mask 255.0.0.0
```

Here's a little explanation for some options:

- The `fudge 127.127.1.10 stratum 10` directive is a “dummy” server acting as fallback IP in case the external time source becomes momentarily unreachable. When this happens, NTP will continue to work and base itself on this “internal” server.
- NTP has its own arsenal of rules to limit access to the service, which can be used independently from a firewall. The `restrict` directives in the above configuration prevent distant computers from changing the servers' configuration (first `restrict` statement), and the machine is configured to trust itself (second `restrict` statement).
- A `restrict` statement without any argument but followed by the hostname boils down to an `allow all`.

Manage the NTP service

Before starting the service, proceed to an initial adjustment of your system clock:

```
# ntpdate pool.ntp.org
```



The `ntpdate` command is normally considered obsolete, but it still comes in handy when performing important time adjustments. The “orthodox” way would be to use the `ntpdate -g` command - the official replacement for `ntpdate` - but its use will fail if your system clock is off for more than half an hour.

Activate the NTP service:

```
# chmod +x /etc/rc.d/rc.ntpd
```

Manage the NTP service:

```
# /etc/rc.d/rc.ntpd start|stop|restart|status
```

Now display the list of servers your machine is actually connected to:

```
# ntpq -p
remote                refid                st t when poll reach  delay  offset jitter
=====
==
*panopea.unstabl 213.251.128.249  2 u  30  64  377  56.136  -249.48
80.680
+88-190-17-126.r 145.238.203.14   2 u  29  64  377  77.571  -205.94
94.278
+62.210.255.117 192.93.2.20      2 u  29  64  377  77.097  -249.57
85.641
-ntp.univ-poitie 145.238.203.10   3 u  29  64  377  57.747  -191.58
107.002
LOCAL(0)         .LOCL.           10 l 164  64  374  0.000   0.000 0.001
```

The little * asterisk preceding one of the above lines means your machine is effectively synchronized with the respective NTP server.



The first synchronization can take a few minutes, sometimes up to half an hour.

Synchronize your client PC(s) with your local NTP server

In a LAN, it is considered good practice to synchronize only one machine - the server - with a public NTP server, and the client PCs with the local server. This saves bandwidth and takes some load off the public NTP servers.

As above, proceed to an initial adjustment of the system clock:

```
# ntpdate pool.ntp.org
```

Create an empty logfile:

```
# touch /var/log/ntp.log
```

Now configure NTP to synchronize with the LAN server. Replace the example's IP (192.168.2.1) with your real server's IP:

```
# /etc/ntp.conf

driftfile /etc/ntp/drift
logfile /var/log/ntp.log

server 192.168.2.1
```

```
server 127.127.1.0
fudge 127.127.1.0 stratum 10

restrict default ignore
restrict 127.0.0.1 mask 255.0.0.0
restrict 192.168.2.1 mask 255.255.255.255
```


- The three restrict statements mean we're blocking all NTP traffic except for the client itself and the server.

Activate and start the NTP service:

```
# chmod +x /etc/rc.d/rc.ntpd
# /etc/rc.d/rc.ntpd start
```

As above, use the ntpq -p command to check if the synchronization went well:

```
# ntpq -p
      remote           refid      st t when poll reach  delay  offset
jitter
=====
==
*192.168.2.1      81.19.16.225    3 u  916 1024  377   0.367   7.897
2.552
LOCAL(0)         .LOCL.          10 l  10h  64    0   0.000   0.000
0.000
```

 Again, you might have to wait a few minutes before the first synchronization takes place.

Monitor the performance of ntpd

You will notice that the logfile /var/log/ntp.log does not contain any information about the actual accuracy of your system clock. If it's important to you, you can log the statistics of time corrections applied by NTP daemon to the system clock. To do this, add the following lines to /etc/ntp.conf:

```
statsdir /var/log/ntp/
statistics loopstats
filegen loopstats file loops type day link enable
```

You have to create the statsdir manually. Once the configuration changes are in effect, ntpd will create files named loops.YYYYMMDD in that directory. Below is an example line from one of these files:

```
56690 3950.569 0.001199636 2.297 0.001830770 0.571576 10
```

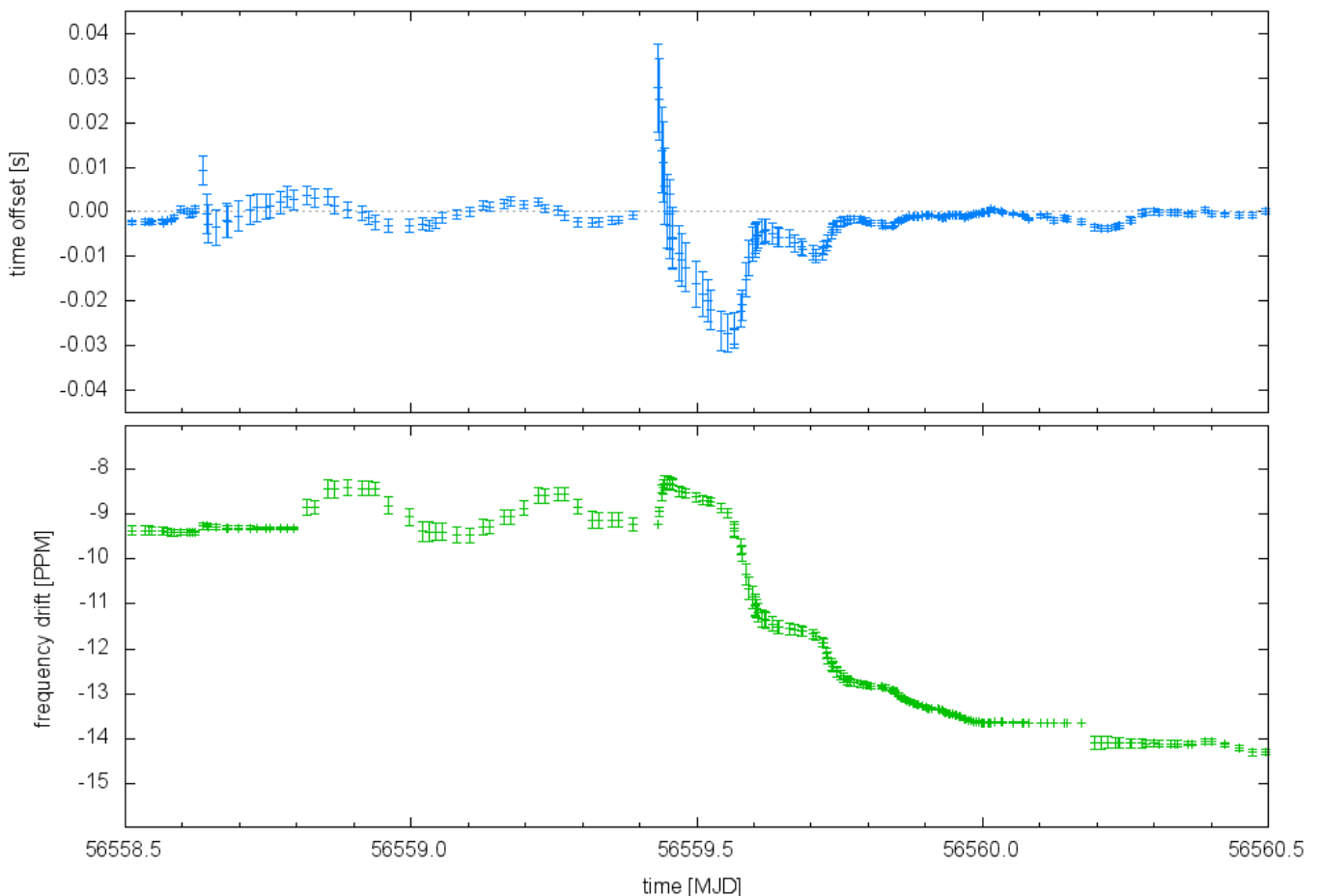
The first and second number are the UTC time (expressed as [Modified Julian Date](#) and seconds

elapsed since midnight). The third and fourth number are the offsets of time (in seconds) and of frequency (in parts per million). The fifth and sixth number are their respective uncertainties.

To monitor the performance of `ntpd`, you can examine the plot of clock offset or frequency offset vs. time:

```
$ awk '{printf("%f %f %f\n", $1+$2/86400, $3, $5)}' /var/log/ntp/loops.* >
time
$ awk '{printf("%f %f %f\n", $1+$2/86400, $4, $6)}' /var/log/ntp/loops.* >
freq
$ gnuplot
gnuplot> set xzeroaxis
gnuplot> plot 'time' with yerror
gnuplot> plot 'freq' with yerror
```

Given enough data, visual examination of the plots will allow you to see peculiarities in `ntpd` performance, should they arise. For example, in the case illustrated by the figure below, the rapid decrease of the frequency offset was caused by replacing the power supply unit of the machine.



Sources

- Originally written by [Niki Kovacs](#)
- Performance monitoring section contributed by Dominik Drobek

[howtos](#), [time](#), [clock](#), [synchronization](#), [author kikinovak](#)

From:
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:
https://docs.slackware.com/howtos:network_services:ntp

Last update: **2014/02/17 21:13 (UTC)**

