

OpenNTPD

OpenNTPD is a small, free implementation of the Network Time Protocol. It is an alternative to the reference implementation available directly from ntp.org.

Advantages (+) and Disadvantages (-) of OpenNTPD:

- + much smaller and lighter weight. (~50k vs. ~400k)
- + Development emphasis on security
- + Simple configuration
- Less flexible
- Not compatible with ntpq

It is recommended that you know how to:

- ... use the command line
- ... install packages using ipkg
- ... set up jffs

Contents

- 1 Pre-Installation Setup
 - ◆ 1.1 /opt
- 2 Installation
- 3 ntp user
 - ◆ 3.1 ntp user directory
- 4 Configuration
- 5 Calibration
- 6 Starting OpenNTPD
 - ◆ 6.1 Editing the ntpd Init Script
 - ◆ 6.2 Using the ntpd Init Script
- 7 Setting the Time Zone
 - ◆ 7.1 httpd
 - ◆ 7.2 cron
- 8 System Startup Scripts
 - ◆ 8.1 Setting TZ globally
 - ◆ 8.2 Copying the clock calibration
 - ◆ 8.3 Updating system files
 - ◆ 8.4 Automatically starting ntpd
 - ◆ 8.5 Restarting system services
 - ◆ 8.6 Disabling the default NTP Client
- 9 External Links

Pre-Installation Setup

Before you install ntpd you need to decide where to install it. Your basic choices are volatile or non-volatile memory (ram or jffs), or an external store (usb or networked). Realize anything you install to ram will be lost once your system reboots.

/opt

Ipkg can be configured to work with any of these, but it is much simpler and much cleaner to always install to /opt and use the mount system to direct /opt to whatever your choice. In this example, packages will be installed to jffs:

```
mkdir /jffs/opt
mount --bind /jffs/opt /opt
```

Installation

First, you will need to find an OpenNTPD package to install. I found [this one](#).

```
export IPKG_INSTROOT=/opt
ipkg -d /opt install http://downloads.openwrt.org/kamikaze/8.09.2/brcm47xx/packages/openntpd\_3.9p1-2\_mipsel.ipk
```

If you prefer you can download the file first, then install it:

```
cd
wget http://downloads.openwrt.org/kamikaze/8.09.2/brcm47xx/packages/openntpd\_3.9p1-2\_mipsel.ipk
ipkg -d /opt ./openntpd_3.9p1-2_mipsel.ipk
```

You can also just use the standard ipkg commands:

```
ipkg update
ipkg -d /opt openntpd
```

Note the update step downloads about 500k of meta-data to your jffs.

ntp user

One of the things the openntpd daemon does to promote security is to split into two processes, one privileged and one not. The process that is not privileged will run as the "ntp" user, which probably does not exist in your /etc/passwd file. If you set IPKG_INSTROOT as above, installation will create two files for you: /opt/etc/passwd and /opt/etc/group. You only need to append these files to their corresponding files in /etc.

```
for i in /etc/passwd /etc/group; do
  cat /opt/$i >>$i
done
```

If you didn't set IPKG_INSTROOT, the above files will already be appended in /etc. The ntp user will have to be created every time your system is rebooted. With the passwd and group file in /opt/etc, it is a simple matter

OpenNTPD

to append these two files to their /etc counterparts. See Startup, below.

ntp user directory

To run as the ntp user, the user's home directory must exist. If you are not using the init file to start ntpd, you will have to manually create the ntp home directory. You will need to do this every time you reboot.

```
mkdir -p `awk -F: '/^ntp:/{print $6}' /etc/passwd`
```

It is not a mistake that we are referring to /etc/passwd and not /opt/etc/passwd since in the previous step we combined the two files.

Configuration

If you do not intend to use your router as a time server, the default ntpd.conf file installed by ipkg is good enough. You can find this file in /opt/etc/ntpd.conf. Otherwise you will need to edit the config file to tell it what port to listen on.

```
listen on 0.0.0.0
servers us.pool.ntp.org
```

If you do not live in the United States, you probably want to use a time server closer to you.

Calibration

OpenNTPD will gradually calibrate your router's clock to match that of the NTP server. Synchronization is very slow, however, and is limited to about 150 microseconds up or down. Since my router's clock was way more off than that (about 7s/hr) I needed to do a rough calibration first by hand. You will need a stopwatch and an accurate time standard.

Using a web browser, connect to your router's home page by typing your router's IP address in the address bar. This will bring up your router's status page. In small print at the upper right of the page, the router's status is displayed. Wait for the time to change and start your stopwatch. Record the current time reported on the status page. Wait at least a half hour and repeat the process: wait for the time to change, stop your stopwatch, and record the reported time. Now divide the time change reported by your router by the actual elapsed time. This should give you a number near 1. Multiply the result by 10,000 and round to the nearest integer. (In my case, the answer was 9981.) Use adjtimex to do a coarse adjustment of your router's system clock:

```
adjtimex -t 9981
```

(Substitute your own computed value for 9981.)

Starting OpenNTPD

Editing the ntpd Init Script

Later we are going to need to use the startup script provided by ipkg in the /opt/etc/init.d directory. Depending on the version of OpenNTPD you downloaded, you may need to edit this file with the correct paths. If you are installing the same version of OpenNTPD as I am you will need to change line 1 in the file to:

```
#!/bin/sh /opt/etc/rc.common
```

and line 13 to read in the config file from /opt:

```
${IPKG_INSTROOT}/usr/sbin/ntpd -sf ${IPKG_INSTROOT}/etc/ntpd.conf
```

Note this makes use of the IPKG_INSTROOT environment variable we set in an earlier step.

In my case I also needed to download the files rc.common and functions.sh into /opt/etc. I found copies I could use at the following locations. Cut and paste the contents into an editor and then delete the line numbers by hand:

<https://dev.openwrt.org/browser/trunk/package/base-files/files/etc/rc.common>
<https://dev.openwrt.org/browser/trunk/package/base-files/files/etc/functions.sh>

Make sure you save the files in /opt/etc.

Using the ntpd Init Script

You can now use the init script to start OpenNTPD. This is preferable to starting NTPD by hand because the init script will check for and create needed directories.

```
/opt/etc/init.d/ntpd start
```

Setting the Time Zone

Confirm ntpd is running using ps.

```
ps
```

You should see two lines similar to the following near the bottom of you listing:

```
12606 ntp      1156 S    /opt/usr/sbin/ntpd -sf /jffs/etc/ntpd.conf
12608 root      1144 S    /opt/usr/sbin/ntpd -sf /jffs/etc/ntpd.conf
```

Check your router's time:

```
date
```

You will probably find that your date is way off. The seconds should be correct, and probably the minutes as well, but the hours will not. Looking at the output of date the reason is clear:

```
Thu Jul  1 14:13:00 UTC 2010
```

OpenNTPD

Note the timezone listed is UTC. dd-wrt thinks the system clock is set to the local time but OpenNTPD sets it to UTC. Fortunately, dd-wrt uses the uclibc library which uses the TZ environment variable to understand the time. Look up the appropriate setting for your timezone [here](#). My timezone is US/Pacific.

```
export TZ="PST8PDT,M3.2.0/2,M11.1.0/2"
```

Now if you run date it should report your time correctly.

```
Thu Jul 1 07:20:04 PDT 2010
```

This only affects your current shell, however. If you go to your internet browser and check the time reported on your router's status page you will see it is still reporting UTC. That is because your TZ environment variable only applies to the current shell and to any processes you start after the variable has been set.

httpd

To update the time on your web page you just need to restart your httpd daemon.

```
killall httpd
httpd -p 80
httpd -S
```

Now your web server should report the correct time.

cron

You probably want cron to use local time as well.

```
killall cron
cron
```

System Startup Scripts

None of this is of any use if you have to manually configure OpenNTPD to run every time your system boots.

Setting TZ globally

To set the TZ environment variable globally, you just need to edit /etc/profile, which is read only. [Useful Scripts](#) shows a trick to make /etc/profile writable.

First, create /jffs/etc/profile and put the following contents in it.

```
./tmp/etc/profile
export TZ="PST8PDT,M3.2.0/2,M11.1.0/2" IPKG_INSTROOT=/opt
```

OpenNTPD

Then add the following to the beginning of your system startup script. The easiest way to do this is via the administration->commands screen from the web configuration utility.

```
cp /etc/profile /tmp/etc/profile
mount --bind /jffs/etc/profile /etc/profile
```

Copying the clock calibration

Use adjtimex to update the system clock calibration. You will need to use the adjtimex coarse calibration done above. Add the following to your system startup script:

```
adjtimex -t 9981
```

Substitute your own calibration value for 9981. If you are using an older OpenNTPD, you will also have to set up the frequency. You can do this by running the command adjtimex at the command prompt after ntpd has been running for a few days to see what the OpenNTPD has come up with. Note that the -t setting will not have changed. The only relevant setting will be the -f (frequency) setting. If you are using an OpenNTPD that does not create /var/db/ntpd.drift, this is what you will have to do.

```
adjtimex -t 9981 -f 1531248
```

Substitute your own value for -f from what adjtimex reported.

If you are using a newer OpenNTPD that writes /var/db/ntpd.drift you can just copy that file to /opt/var/db/ntpd.drift after OpenNTPD has been running a few days. The following step will automatically copy it to /var/db when the system is rebooted.

Updating system files

Since the passwd and group files get reset whenever the system is rebooted, you will need to add the ntp in your system startup script:

```
for i in /etc/passwd /etc/group /var/db/ntpd.drift; do
  [ -r ${IPKG_INSTROOT}${i} ] && mkdir -p $(dirname $i) && cat ${IPKG_INSTROOT}${i} >>${i}
done
```

Automatically starting ntpd

dd-wrt will start scripts automatically that are in the /etc/init.d directory. It does this by running /etc/init.d/rcS. Unfortunately, rcS and /etc/init.d are not writable, so we will have to use the mount --bind trick again to change that.

```
cp -rp /etc/init.d /opt/etc/init.d
```

For the version of OpenNTPD I downloaded, I had to edit the rcS script to use the rc.d directory instead of the init.d directory.

```
#!/bin/sh
```

OpenNTPD

```
for i in /opt/etc/rc.d/S*; do
  $i start 2>&1
done | logger -s -p 6 -t &
```

To add ntpd to rc.d I just ran

```
/opt/etc/init.d/ntpd enable
```

For earlier versions you will just need to make sure there is a file S60ntpd in your /opt/etc/init.d directory. You can make it a symbolic link to the ntpd file:

```
ln -s ntpd /opt/etc/init.d/S60ntpd
```

Now you just need to override /etc/init.d with the directory we just created using mount --bind. Append the following to your system startup script (via the web configuration interface):

```
mount --bind /jffs/opt /opt
mount --bind /opt/etc/init.d /etc/init.d
```

Note: For v24-sp2 at least, /etc/init.d/rcS is not called during startup. In that case there is no purpose mounting /opt/etc/init.d over the default /etc/init.d. Instead, call your edited rcS from your own startup script.

```
/opt/etc/init.d/rcS
```

Restarting system services

Many system services ignore the default /etc/profile and need to be restarted to take advantage of the updated time zone. If, after rebooting the time reported by your router's status page is not correct for your time zone, you will need to restart at least httpd and cron to set the time zone properly:

```
for i in httpd cron; do
  /sbin/stopservice $i && /sbin/startservice $i
done
```

You can add services to this list as needed.

Disabling the default NTP Client

Don't forget to disable dd-wrt's native NTP client at the bottom of the setup page in the web configuration tool.

External Links