www.internet2.edu

Network Time Protocol (NTP) Overview and Configuration: An Internet2 Cookbook

Disclosure/Disclaimer

This document is based on a talk presented by Stanislav Shalunov and is designed to be used in conjunction with an Internet2 Network Performance Workshop; for more information on these workshops (upcoming and past), see: http://e2epi.internet2.edu/network-perf-wk/.

Cookbook Contents

This cookbook will cover an overview of NTP and instructions on installing ntpd. The NTP Overview portion will include sections on:

- Purpose and utility
- Strata, the loosely coupled network
- Limitations, need for symmetry– Influence of temperature
- Network use
- Use of cryptography
- Implementations of NTP

Installing ntpd includes sections on OS selection, obtaining, compiling, and installing ntpd, configuring ntpd, and monitoring ntpd.

NTP Overview

This section will cover the purpose and utility of NTP, packet exchange, strata, limitations, topological symmetry, influence of time and temperature, network use, use of cryptography, and various implementations.

Purpose of NTP

The main purpose of NTP is to synchronize clocks of a server constellation to UTC. It does not work very well for synchronizing to a common time with no external reference. What can NTP do?

1) Minimize offset (difference from the true time) and 2) minimize skew (difference of time change rate from the true rate). Realize that, sometimes, these goals contradict each other; NTP compromises, sensibly. Note: NTP does not deal with time zones (that’s a time display problem).

Utility of NTP

A great part of the utility of NTP is that timestamps (e.g., in log files) make sense; timestamps can be compared to those made on other machines. Many tools that depend on time work in distributed environments when NTP is used. Specifically:

- `make` (the UNIX dependency builder) on an NFS-mounted file system
- Kerberos
- SecurID tokens
- Since NTP minimizes skew, time stepping (which breaks all of the above) is avoided.

In addition, one-way delay can be measured; one-way delay measurement requires better time than many (most?) other applications.

Packet Exchange of NTP

NTP uses UDP port 123 and its own binary format (RFC 1305, RFC 2030). (There is no public online version of the NTPv4 specification at this time.) By exchanging packets, the client can estimate the offset between it and the server. The client can have several servers but choose one server to synchronize to at any given time. Once a client selects its server, it minimizes the offset and skew with a feedback loop.

The client sends request with a timestamp; the server returns a packet with three timestamps:

1. Echo of the client timestamp
2. When did I receive the request?
3. When did I send the response?

NTP: Strata

Each NTP node has a stratum. Stratum is an integer between 0 and 16, inclusively; stratum 0 means a physical clock, never a computer. Examples of physical clocks include:

- Cesium oscillator: Definition of time (subject to relativistic effects)
- Rubidium oscillator: found in cell towers, very stable
- GPS receiver: accuracy circa 10 ns
- CDMA receiver: accuracy circa 10 µs

Stratum 16 is reserved for devices that are not synchronized. The stratum of any NTP-synchronized device is the stratum of the device it is synchronized to, plus 1. Thus:

- GPS receiver: stratum 0
- Computer connected to it by a serial line: stratum 1
- Client that gets the time from that computer: stratum 2

The NTP Network

NTP servers form a loosely coupled network. Each node decides which server to use for synchronization based on complex selection algorithm (voting-like). The selection algorithm runs repeatedly for protection against falsetickers. By virtue of having the complex selection algorithms, NTP:

- runs sanity checks/error estimates,
- has resiliency to clock and network failure, and
- enforces the global nature of NTP.

Limitations

NTP needs external servers to work well, even with a local clock. It can produce systematic errors with asymmetric paths and can have problems with asymmetric congestion. (And no time zones, remember?)

Topological Symmetry

NTP needs topological symmetry. Assume asymmetric paths between client and server; packet needs t^1 to get from client to server and t^2 from server to client. The offset that the client will see is the true offset plus:

$$\frac{t^1 - t^2}{2}$$

There is no way around this: the extra offset shows up as an error in the server clock. If there are multiple servers, not is all lost! However, the paths must be diverse, or all the servers will have the same error.

Time and Temperature

The oscillator frequency depends on temperature. A typical correspondence is 1PPM (part per million) of clock rate for 1ºC. (NTP can resolve rate differences of .001PPM.) For comparison, the temperature inside a modern computer will vary by 10ºC, depending on CPU load – NTP could notice human movement around the host (!), and certainly open windows or A/C failures. NTP will compensate for frequency variation, but the best servers sit in constant-temperature machine rooms.

Network Use by NTP

NTP requires a pair of small packets every 64–1024 seconds for each server, with eight times more if `burst` keyword is used. For the vast majority of sites, this is nothing to worry about. (It represents less than 0.01% of Abilene traffic in bytes [circa 5 GB/day for the entire R&E world]. For comparison: DNS is circa 0.15% of Abilene traffic in bytes.)

Cryptographic Authentication

The motivation for this new feature is to allow clients to authenticate time sources. It is not present in NTPv3 and relatively new in NTPv4. It is not needed for the vast majority of cases; however, no cryptography will help against an attacker who delays packets. This feature only makes sense for time distribution within organization. That said, only the shared keys part is likely to work at this time. The question you might ask yourself is: Does your ultimate time source (e.g., GPS receiver) get its time through a cryptographically authenticated channel? Current best advice: do not bother with cryptography for NTP.

Implementations

The old utility, xntpd (shipped with some ancient operating systems), implements NTPv3 and comes from the University of Delaware. The new ntpd is the current state of the art and also comes from the University of Delaware. There are no complete NTP implementations for Windows (SNTP only, numerous). OpenNTPD is not an NTP implementation. If you were interested enough to read this far, you want ntpd.

Installing ntpd

This section will cover:

- OS selection for ntpd
- Hardware Requirements
- Obtaining ntpd
- Compiling/installing ntpd
- Enabling ntpd

OS Selection for ntpd

David Mills, author of the NTP standards, xntpd, and ntpd, prefers FreeBSD; a few quotes:

"I have confronted Linux many, many times with broken this and that and concluded I will not attempt to examine, fix, or judge anything Linux. You are on your own." —David L. Mills, 2004-10-20

"There have been reports that Linux is unfriendly at clock frequencies other than 100 Hz." —David L. Mills, 2004-05-07

"I suspect the Linux team has not noticed that the adjtime() and ntp adjtime() syscalls needs to recalibrate the tick adjustment value so to achieve something like 500 PPM as with 100Hz clocks. [...] Unless Linux surmounts that challenge, NTP is a lose." —David L. Mills, 2004-03-23

"I say this in the urgent agenda to avoid Linux recursive complexity and keep FreeBSD clean and clear of messing fingers." —David L. Mills, 2004-02-16

"[Y]ou did say Linux. All of our tests are in other systems and they work as intended." —David L. Mills, 2003-05-31

"When you said 'Linux' the red light came on." —David L. Mills, 2003-05-11

"I am even more astonished if anybody considers Linux NTP anywhere near a conformant implementation of anything." —David L. Mills, 2003-04-10

"The TAI [code] works perfectly in FreeBSD with ntpd and ntp-time and Autokey. Linux experience may be different." —David L. Mills, 2003-03-24

Und so weiter, und so fort, again and again ad infinitum. However:

"Most of you know my feelings about Linux, and I am on record as not allowing Linux anywhere near here, but the Linux system we don't have runs 2.6.4 kernel and now running latest ntpd 4.2.0. With virtually no customization (NFS and NIS and ssh) and vanilla ntp.conf it runs like a charm with or without the kernel [NTP support] enabled." —David L. Mills, 2004-04-01

Thus, if convenient, use FreeBSD in favor of Linux. If you want stratum 1, FreeBSD is strongly recommended (Linux needs non-standard kernel mods, PPSkit, to run well as stratum 1). Note that Linux should work, but distributions can have surprises. Other Unix systems might work, even if not as well (including Mac OS X), but don't even think about Windows.

Hardware Requirements for NTP

The actual hardware requirements are simple: a CPU and a network interface. Memory and hard disk are optional – an old 486 will handle the load of a department. The big issue here is temperature stability (lack of advanced power management and HLT instruction); it is more important than the rest. Anything, really, will work but you need a time source and good interrupt handling ability for stratum 1. NTP appliances make a decent stratum 1.

Obtaining ntpd

On FreeBSD, use the system binaries. On other OSes, especially Linux, the system packager/distributor/vendor might have broken ntpd. Uninstall the system ntpd and grab the latest production tarball from http://www.ntp.org/.

Compiling/Installing ntpd

On FreeBSD, do nothing. On other OSes, ensure OpenSSL is installed, if you want NTP cryptography. Then,

```
tar xzf ntp4.2.0.tar.gz
cd ntp4.2.0
./configure
gmake
su
gmake install
```

(Change 4.2.0 to the appropriate version number, of course.)

Enabling ntpd

On FreeBSD, insert the line "`ntpdenable="YES"`" into `/etc/rc.conf`. On Linux, the official way depends on the distribution, but it's probably easiest to add the ntpd command to the end of `/etc/rc.local`.

Configuring ntpd

This section covers:

- Configuration file location and format
- `broadcastclient` and `multicastclient`
- Selecting the stratum servers
- Selecting the number of servers
- Server strata
- Server location
- Useful server options Access Control

Configuration File Location and Format

The config file is `/etc/ntp.conf`. It consists of a sequence of statements, one configuration statement per line. The most important statements are:

```
      driftfile <filename>
      server <ntp.example.edu> [<options>]
```

It is important to include `driftfile <filename>`; note that it is not on by default. On FreeBSD, `driftfile` is normally `/var/db/ntp.drift`. On Linux, it varies, but there's `/etc/ntp/drift` in most cases. The historic location is `/etc/ntp.drift`.

These two statements are enough to make a working server.

{broad,multi}castclient

This is a lazy way to configure – `broadcastclient` will listen to broadcasts and `multicastclient <IP address>` will listen to multicast on 224.0.1.1 (or another). It can be used for end-user machines but ***do not use on machines that will run OWAMP***.

Selecting the Stratum

Choose stratum 1 if you need maximum possible accuracy (single microseconds); stratum 2 is suitable for machines that serve time to others or are used in measurements (accuracy can be better than 1 ms). Stratum 3 is a reasonable end-user stratum.

Most measurement nodes will probably use NTP stratum 2 because it provides the best accuracy 'bang for the buck;' sufficient accuracy for most needs and can be set up on most machines with no new hardware. The best measurement nodes will use stratum 1.

These nodes will need a stratum 0 time source. Oscillators are too expensive for most users, but, you can choose between CDMA receivers, which are cheap and work anywhere a CDMA cell phone works and GPS receivers, which provide the best accuracy "bang for the buck," but require an antenna that can see the sky.

Note: Leave stratum 3 for your laptop and (maybe) your mail server

Stratum 1 Servers

Stratum 1 servers provide the best accuracy but require special hardware, extra work in configuration, and should really run FreeBSD. They should use pulse per second (PPS) mode instead of polling mode.

Selecting the Number of Servers

Lazy people select one server (perhaps pool.ntp.org) – Don't be lazy! Time can be OK with one server, but error estimate is worthless. The minimum number of servers for a reasonable error estimate is 3. The minimum number of server to detect one falseticker is 4. Anything in excess of 6 servers is likely to result in too many switches from server to server.

Prepare for losing connectivity to some servers (at least you should be able to survive single server failure). Reasonable numbers are 4 well-selected servers *or* 5 or 6 servers selected with, perhaps, not the same degree of care.

Server Strata

Only servers of stratum n-1 (where n is my stratum) fully matter. It is important to select servers of the same strata. Use all stratum 1 peers for Stratum 1 configurations (unless you can have multiple physical time sources) and use all stratum 1 servers for Stratum 2 configurations. Use all stratum 2 servers for Stratum 3 configurations (but don't do stratum 3 configurations for measurement). Avoid mixing strata.

Server Location

Good servers are topologically close; the best server is your own server (get a GPS). The second-best server is your provider's server: demand it.

GigaPoPs can use the closer of the ntp-e.abilene.ucaid.edu and ntp-w.abilene.ucaid.edu for stratum 2; contact us for stratum 1 information. The third-best server is your neighbor's server; ask around.

If you still haven't found 4 or 5 servers, there's the public server list at http://ntp.isc.org/bin/view/Servers/StratumOneTimeServers. Hint: servers operated by NIST and USNO are well-administered and tend not to go away (you pay for them with your tax dollars, might as well use them).

Useful Server Options

The one I find most useful is `iburst`. It allows you to send more packets in the beginning and synchronize faster. One might also consider `burst` (more packets, better accuracy) and `tinker huffpuff 7200`, which can be a big help in case of asymmetric congestion (but not asymmetric topology).

Access Control

Some people like to restrict access so that only the servers can talk to them. This is usually an overkill; theft of NTP service is uncommon. Such restrictions are difficult to administer: when servers change IP addresses, you need to change the config file.

Monitoring ntpd

This section covers `ntpq –p`, `ntptime`, and statistics (`{loop,peer,clock,raw}stats`).

ntpq -p

Outputs a two-line header and then one line per server, with eleven fields:

1. First character: '*' for the host we sync to; '+' for OK; '—' and 'x' for bad; ' ' for unreachable; 'o' for PPS sync source
2. Hostname
3. Where does it get time from?

4. Its stratum
5. Type of peer (‘u’ is for ‘unicast’)
6. Seconds ago we heard from it
7. How often do we currently poll, in seconds
8. Octal reachability mask, 377 means A-OK
9. Delay (round-trip) to the server, in milliseconds
10. Offset in milliseconds
11. Jitter in milliseconds

Here’s an example of `ntpq –p` output for a mixed-strata config:

```
$ ntpq –p mail.internet2.edu
Remote            refid              st   t   when   poll  reach   delay   offset  jitter
================  =================  ==   =   ====   ====  =====   =====   ======  ======
-ntp0.usno.navy.  .USNO.             1    u   651    1024  33      78.347  18.023  0.124
+ntp-e.abilene.u  nms4-nycm.abilene  2    u   238    1024  377     34.944  0.734   0.143
+ntp-w.abilene.u  Nms4-snva.abilene  2    u   77     1024  377     55.667  0.402   0.137
207.75.164.25     0.0.0.0            16   u   -      64    0       0.000   0.000   4000.0
*pgs1.tns.its.ps  .GPS.              1    u   152    1024  377     52.301  0.445   0.040
```

ntptime

This is the simplest way to look at NTP state; you can learn if NTP is synchronized, maximum error (not very useful), estimated error, the offset from target, and the frequency skew in PPM.

Statistics {loop,peer,clock,raw} Stats

This is a way to really understand what is going on – this is where open window detection comes in. Outside the scope of this presentation.

Resources

```
man ntpd
man ntp.conf
/usr/share/doc/ntp/ (complete HTML documentation)
e2epi.internet2.edu/owamp/details.html#NTP
twiki.ntp.org/bin/view/Support/SelectingOffsiteNTPServers
```

Newsgroup: comp.protocols.time.ntp (free tier 4 support: David Mills might answer your question here—unless you’re using Linux, of course).

Summary of Minimum Working Configuration

- Decide on stratum 2
- Install latest FreeBSD release on an old box
- Select four stratum 1 servers, or see http://www.internet2.edu/˜shalunov/ntp-conf/
- Edit `/etc/ntp.conf`
- Type “ntpd” as root
- Wait a day or so for the clock to settle
- Try to keep the temperature constant