



Aaron Toponce

{ 2013.11.05 }

Real Life NTP

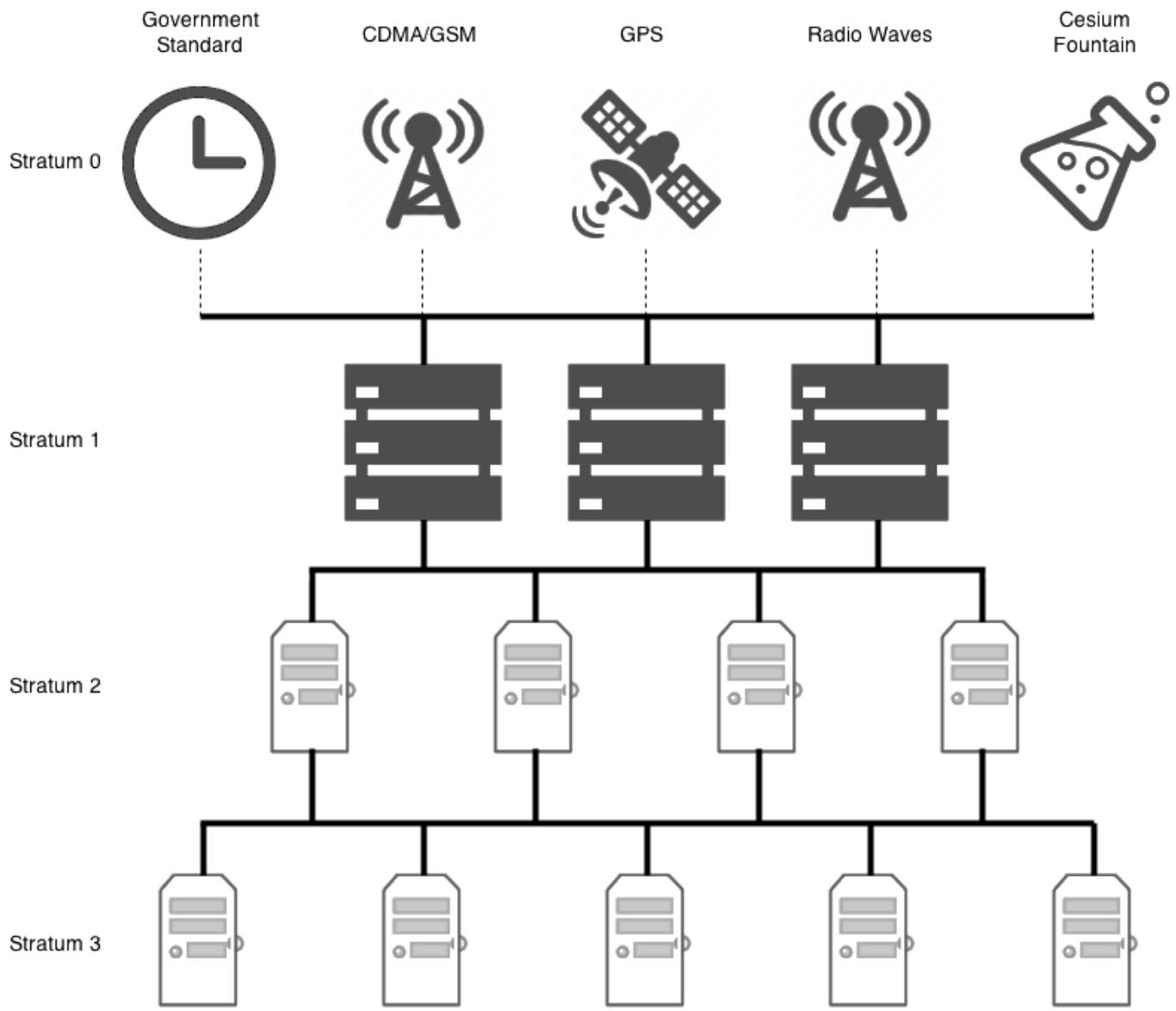
I've been spending a good amount of my spare time recently configuring NTP, reading the documentation, setting up both a stratum 1 and stratum 2 NTP server, and in general, just playing around with NTP. This post is meant to be a set of notes of what I've learned in the process, and hopefully, it can benefit you. It's not meant to be an exhaustive, or authoritative set of instructions on how you should configure your own NTP installation.

Strata

Before getting into the client configuration, we need to understand how NTP serves time to clients. We need to understand the concept of "strata" or "stratum". An authoritative time source, such as GPS satellites, cesium atomic fountains, WWVB radio waves, and so forth, are referred to as "stratum 0" clocks. They are authoritative, because they have some way of maintaining extremely accurate timekeeping. Any time source will suffice, including a standard quartz oscillating clock. However, knowing that quartz based clocks can gain or lose up to 15 seconds per month, we don't generally use them as time sources. Instead, we're interested in time sources that don't gain or lose a second in 300,000 years, as an example.

Computers that connect to these accurate time sources to set their local time are referred to as "stratum 1" time sources. Because there is some inherent latencies involved with connecting to the stratum 0 time source, and the latencies involved with setting the time, as well as the drift that the stratum 1 clocks will exhibit, these stratum 1 computers may not be as accurate as their stratum 0 neighbors. In real life, the clocks on good stratum 1 computers will probably drift enough that their time will be off by a couple microseconds, compared to the stratum 0 source that they are getting their time.

Computers that connect to stratum 1 computers to synchronize their clocks are referred to as "stratum 2" time sources. Again, due to many latencies involved, stratum 2 clocks may not be as accurate as their stratum 1 neighbors, and even worse compared to the further upstream stratum 0 time sources. In practice, your stratum 2 server will probably be off from its stratum 1 upstream server by anywhere from a few microseconds to a few milliseconds. Many factors come into play in how this is calculated, but realize that stratum 2 computers, in practice, are probably the furthest time source from stratum 0 that you want to synchronize your clocks with.



As you would expect, stratum 3 clocks are connected upstream to stratum 2 clocks. Stratum 4 clocks are connected upstream to stratum 3 clocks, and so forth. Once you reach the lowest level of stratum 16, the clock is now considered to be unsynchronized. So again, in practice, you probably don't want to sync your computers clock with any strata lower than 2, thus making your computer a stratum 3. At this point, you're far enough away from the "true time" source, that your computer could exhibit time offsets anywhere from a few milliseconds to several hundred milliseconds.

If your clock is off by 1000 seconds, NTP will refuse to synchronize your clock, and it will require manual intervention. If the upstream stratum from which you are synchronizing your clock is off by 1000 milliseconds, or 1 full second, that time source will not be used in synchronizing your clock, and others will be picked instead (this is to help weed out bad time sources).

Client

Debian, Ubuntu, Fedora, CentOS, and most operating system vendors, don't package NTP into client and server packages separately. When you install NTP, you've made your computer both a server, and a client simultaneously. If you don't want to serve NTP to the network, then don't open the port in your firewall. In this section, we'll assume that you're not going to use NTP as a server, but wish to use it as a client instead.

I'm not going to cover everything in the `/etc/ntp.conf` configuration file, which is generally the standard installation path. However, there are a few things I do want to cover. First, the "server" lines. You can have multiple server lines in for configuration file. NTP will actively use up to 10. However, how many do you add? Consider the following:

1. If you only have one server configured, and that server begins to drift, then you will blindly follow the drift. If that server consistently gained 5 seconds every month, so would you.
2. If you only have two servers configured, then both will be automatically assigned as "false tickers" by NTP. If one of the servers began to drift, NTP would not be able to tell which upstream server is correct, as there would not be a quorum.
3. If you have three or more servers configured, then you can support "false tickers", and still have an agreement on the exact time. If you have five or six servers, then you can support two false tickers. If you have seven or eight servers, you can support three false tickers, and if you have nine or ten servers configured, then you can support up to four false tickers.

NTP Pool Project

As a client, rather than pointing your servers to static IP addresses, you may want to consider using the NTP pool project. Various people all over the world have donated their stratum 1 and stratum 2 servers to the pool, Microsoft, [XMission](#), and [even myself](#) have offered their servers to the project. As such, clients can point their NTP configuration to the pool, which will round robin and load balance which server you will be connecting to.

There are a number of different domains that you can use for the round robin. For example, if you live in the United States, you could use:

- 0.us.pool.ntp.org
- 1.us.pool.ntp.org
- 2.us.pool.ntp.org
- 3.us.pool.ntp.org

There are round robin domains for each continent, minus Antarctica, and for many countries in each of those continents. There are also round robin servers for projects, such as Ubuntu and Debian:

- 0.debian.pool.ntp.org
- 1.debian.pool.ntp.org
- 2.debian.pool.ntp.org
- 3.debian.pool.ntp.org

ntpq(1)

NTP ships with a good client utility for querying NTP; it's the `ntpq(1)` utility. However, understanding the output of this utility, as well as its many subcommands, can be daunting. I'll let you read its manpage and documentation online. I do want to discuss its peering output in this blog post though.

On my public NTP stratum 2 server, I run the following command to see its status:

```
$ ntpq -pn
      remote          refid      st t when poll reach    delay   offset  jitter
=====
*198.60.22.240 .GPS.        1 u  912 1024  377    0.488  -0.016   0.098
+199.104.120.73 .GPS.        1 u   88 1024  377    0.966   0.014   1.379
-155.98.64.225 .GPS.        1 u   74 1024  377    2.782   0.296   0.158
-137.190.2.4   .GPS.        1 u 1020 1024  377    5.248   0.194   0.371
-131.188.3.221 .DCFp.       1 u  952 1024  377  147.806  -3.160   0.198
-217.34.142.19 .LFa.        1 u  885 1024  377  161.499  -8.044   5.839
-184.22.153.11 .WWVB.       1 u  167 1024  377   65.175  -8.151   0.131
+216.218.192.202 .CDMA.      1 u   66 1024  377   39.293   0.003   0.121
-64.147.116.229 .ACTS.       1 u   62 1024  377   16.606   4.206   0.216
```

We need to understand each of the columns, so we understand what this is saying:

- **remote**- The remote server you wish to synchronize your clock with
- **refid**- The upstream stratum to the remote server. For stratum 1 servers, this will be the stratum 0 source.
- **st**- The stratum level, 0 through 16.
- **t**- The type of connection. Can be "u" for unicast or anycast, "b" for broadcast or multicast, "l" for local reference clock, "s" for symmetric peer, "A" for a anycast server, "B" for a broadcast server, or "M" for a multicast server
- **when**- The last time when the server was queried for the time. Default is seconds, or "m" will be displayed for minutes, "h" for hours and "d" for days.

- **poll**- How often the server is queried for the time, with a minimum of 16 seconds to a maximum of 36 hours. It's also displayed as a value from a power of two. Typically, it's between 64 seconds and 1024 seconds.
- **reach**- This is an 8-bit left shift octal value that shows the success and failure rate of communicating with the remote server. Success means the bit is set, failure means the bit is not set. 377 is the highest value.
- **delay**- This value is displayed in milliseconds, and shows the round trip time (RTT) of your computer communicating with the remote server.
- **offset**- This value is displayed in milliseconds, using root mean squares, and shows how far off your clock is from the reported time the server gave you. It can be positive or negative.
- **jitter**- This number is an absolute value in milliseconds, showing the root mean squared deviation of your offsets.

Next to the remote server, you'll notice a single character. This character is referred to as the "tally code", and indicates whether or not NTP is or will be using that remote server in order to synchronize your clock. Here are the possible values:

- " " Discarded as not valid. Could be that you cannot communicate with the remote machine (it's not online), this time source is a ".LOCL." refid time source, it's a high stratum server, or the remote server is using this computer as an NTP server.
- "x" Discarded by the intersection algorithm.
- ":" Discarded by table overflow (not used).
- "-" Discarded by the cluster algorithm.
- "+" Included in the combine algorithm. This is a good candidate if the current server we are synchronizing with is discarded for any reason.
- "#" Good remote server to be used as an alternative backup. This is only shown if you have more than 10 remote servers.
- "*" The current system peer. The computer is using this remote server as its time source to synchronize the clock
- "o" Pulse per second (PPS) peer. This is generally used with GPS time sources, although any time source delivering a PPS will do. This tally code and the previous tally code "*" will not be displayed simultaneously.

Lastly, in understanding the output, we need to understand the what is being used as a reference clock in the "refid" column.

- **IP address**- The IP address of the remote peer or server.
- **.ACST**.- NTP manycast server.
- **.ACTS**.- Automated Computer Time Service clock reference from the American National Institute of Standards and Technology.
- **.AUTH**.- Authentication error.
- **.AUTO**.- Autokey sequence error.
- **.BCST**.- NTP broadcast server.
- **.CHU**.- Shortwave radio receiver from station CHU operating out of Ottawa, Ontario, Canada.
- **.CRYPT**.- Autokey protocol error
- **.DCFx**.- LF radio receiver from station DCF77 operating out of Mainflingen, Germany.
- **.DENY**.- Access denied by server.
- **.GAL**.- European Galileo satellite receiver.
- **.GOES**.- American Geostationary Operational Environmental Satellite receiver.
- **.GPS**.- American Global Positioning System receiver.
- **.HBG**.- LF radio receiver from station HBG operating out of Prangins, Switzerland.
- **.INIT**.- Peer association initialized.
- **.IRIG**.- Inter Range Instrumentation Group time code.
- **.JJY**.- LF radio receiver from station JJY operating out of Mount Otakadoya, near Fukushima, and also on Mount Hagane, located on Kyushu Island, Japan.
- **.LFX**.- Generic LF radio receiver.
- **.LOCL**.- The local clock on the host.
- **.LORC**.- LF radio receiver from Long Range Navigation (LORAN-C) radio beacons.
- **.MCST**.- NTP multicast server.
- **.MSF**.- National clock reference from Anthorn Radio Station near Anthorn, Cumbria.
- **.NIST**.- American National Institute of Standards and Technology clock reference.
- **.PPS**.- Pulse per second clock discipline.
- **.PTB**.- Physikalisch-Technische Bundesanstalt clock reference operating out of Brunswick

- and Berlin, Germany.
- **.RATE.**- NTP polling rate exceeded.
 - **.STEP.**- NTP step time change. The offset is less than 1000 millisecends but more than 125 milliseconds.
 - **.TDF.**- LF radio receiver from station TéléDiffusion de France operating out of Allouis, France.
 - **.TIME.**- NTP association timeout.
 - **.USNO.**- United States Naval Observatory clock reference.
 - **.WWV.**- HF radio receiver from station WWV operating out of Fort Collins, Colorado, United States.
 - **.WWVB.**- LF radio receiver from station WWVB operating out of Fort Collins, Colorado, United States.
 - **.WWVH.**- HF radio receiver from station WWVH operating out of Kekaha, on the island of Kauai in the state of Hawaii, United States.

Client Best Practice

There seem to be a couple long standing myths out there about NTP configuration. The first is that you should only use stratum 1 NTP servers, because they are closest to the true time source. Well, this isn't always the case. Connecting to stratum 1 time servers that have high RTT latencies could exhibit large jitter and large offsets. Rather, you should find stratum 1 servers that are physically close to your client. Also, many stratum 1 servers might be overloaded, and finding less stressed stratum 2 servers might deliver more accurate results.

The other myth out there is that you should only connect to physically close NTP servers. This isn't necessarily true either. If the closest NTP servers to you only have one physical link, and that link goes down, you're sunk. Further, if the closest NTP servers to you are stratum 4 or 5 servers, you may exhibit high offsets from the upstream stratum 0 sources. There is a reason why the NTP Pool Project only lists public stratum 1 and stratum 2 servers, and there's a reason why stratum 16 is considered unsynchronized.

Point is, there is a balance in configuring NTP. If you have a large infrastructure, it would make sense for you to build and install a stratum 1 or stratum 2 source at each logically different location (geographically or VLAN'd), and have each server and workstation connect to that logically local NTP server. If it's just your personal computer, then it probably makes sense to just use the NTP Pool Project, and use the round robin domain names. You should keep efficiency and redundancy in mind.

So, you should probably consider the following best practices when configuring your NTP client:

- Use at least 3 servers, and don't statically use busy servers.
- Consider using the NTP pool project, if you will be operating as a client only.
- If statically setting IP addresses for your servers, try to keep the following in mind:
 - Use servers that are physically close to your computer. These servers should have low ping latencies.
 - Use servers that are geographically separated across the globe. Just in case the trans-Atlantic cable is cut, you can still communicate to other servers.
 - Use servers that use different time sources. If all of your servers use GPS as their time source, and GPS goes offline, you will not have anything to synchronize your clocks against.
 - Consider using all 3 of the above on a single client.

Posted by Aaron Toponce on Tuesday, November 5, 2013,

at 11:52 am. Filed under [General](#), [Linux](#), [Ubuntu](#). Follow any responses to this post with its [comments RSS](#) feed.

You can [post a comment](#) or [trackback](#) from your blog. For IM, Email or Microblogs, here is the [Shortlink](#).

{ 4 } Comments

1. MAS | November 6, 2013 at 6:14 pm | [Permalink](#)

Nice writeup. Debian provides the package ntpdate which is just a client to set the date and time of the machine on which it is run. Per the info:

...
ntpdate is a simple NTP client that sets a system's clock to match the time obtained by communicating with one or more NTP servers. It is not sufficient, however, for maintaining an accurate clock in the long run. ntpdate by itself is useful for occasionally setting the time on machines that do not have full-time network access, such as laptops.

2. [MikeDawg](#) | November 7, 2013 at 9:03 pm | [Permalink](#)

Your article is very informative, however, I would like to point a couple things out (from my reddit post):

Going to go on a short rant here.

Overall, this page was very informative; however, there are a couple blaring problems and/or mistakes.

1) A stratum 1 (2 or 3 or so on) server is only as good as its stratum 0 server. Just because the server is labelled as a stratum 0 server, doesn't mean it necessarily has an excellent time source that it is collecting information from.

Fix> Multiple time servers listed for collecting out time information. With 3 servers, you can potentially find 1 bad clock ("falseticker"). With 7 servers, you can potentially find 2 bad clocks in your entire server list.

2) A stratum 1 .. 4 server isn't as accurate as a stratum 0 server.

Fix> A stratum 1 .. 4 server can be more accurate than a stratum 0 server. Again, the important thing, is to maintain a solid list (I generally recommend 3 - 7) of servers. I also recommend only using stratum 2 or 3 servers. There is no reason to tax stratum 1 or stratum 0 servers more than they already are. I hate having to explain this to senior level engineers/administrator at my work. The key is to have a good, reputable number of servers you can sync time to, NOT making sure you connected to a stratum 0 server. As an administrator, if you can verify that your time source is connected to 7 reputable time sources, with relatively low network latency, then you are going to be extremely accurate. We're talking micro seconds. You know, 1/1000000 of a second. Your RTC is going to waver more than that based upon voltage irregularities in your system.

Engineers/Administrators hear that a stratum 3 server is potentially microseconds different than a stratum 0 server, and they freak out. Many people forget, that a microsecond is 1/1000000 of a second. These engineers hear that, and it immediately goes to their head, that it is minutes apart as far as accuracy is concerned.

NTP servers are smart, and they are generally programmed quite well, they detect latency, drift, and a whole lot of other things that you don't really need to be concerned with; but they are accurate.

In the end, this is purely about quantity over quality (for the most part). I would take seven stratum 3 servers, with an acceptable amount of network latency over a single stratum 0 server, with questionable authority, and questionable network latency.

3. [Aaron Toponce](#) | November 27, 2013 at 9:37 am | [Permalink](#)

1) A stratum 1 (2 or 3 or so on) server is only as good as its stratum 0 server. Just because the server is labelled as a stratum 0 server, doesn't mean it necessarily has an excellent time source that it is collecting information from.

Can you show me where I'm claiming that a stratum 0 is keeping True Time? I thought I made it clear that even stratum 0 clocks drift.

Multiple time servers listed for collecting out time information. With 3 servers, you can potentially find 1 bad clock ("falseticker"). With 7 servers, you can potentially find 2 bad clocks in your entire server list.

Not as I understand it. NTP needs a quorum to set the time. This means a majority. 1 false ticker in 3, 2 in 5, 3 in 7, 4 in 9, etc. The number of false tickers must be less than half of the total sources.

Regarding point #2, I see how I can clarify my post. A stratum 4 server might certainly be more accurate than an authoritative stratum 0 source. I'll adjust the language. I also agree with not taxing stratum 1 time sources. However, in practice, I have found stratum 3 and stratum 4 time sources to exhibit a great deal of unnecessary jitter. Sure, at the moment, a stratum 4 server might be more accurate than a stratum 1, but in practice, I rarely see NTP use lower stratum servers when higher are configured.

Again, I agree with not unnecessarily burdening stratum 1 servers. However, if you have your server listed as "OpenAccess" without notification at <http://support.ntp.org/bin/view/Servers/StratumOneTimeServers>, then I see no problem with finding stratum 1 servers close to you with low latencies, and using them for your time source. I would much prefer to take authoritative, reliable, and low latency servers from this list, then sync with some random stratum 3 and stratum 4 time sources in the NTP pool, especially where they restrict queries from outside sources.

Engineers/Administrators hear that a stratum 3 server is potentially microseconds different than a stratum 0 server, and they freak out. Many people forget, that a microsecond is 1/1000000 of a second. These engineers hear that, and it immediately goes to their head, that it is minutes apart as far as accuracy is concerned.

Sure, but even stratum 1 servers are already a few microseconds off, and stratum 2 servers frequently show several hundred microsecond to dozens of millisecond offsets. I've seen stratum 3 servers with hundreds of milliseconds offsets. At that point, yes, I'm freaking out. I expect my DNS and SQL logs to be spot on to the exact millisecond. I don't have the luxury of +/- 100ms offsets for my logs.

In practice, it's trivially easy to setup a local stratum 1 for your office, and have your workstations and servers get your time from it. You can build a stratum 1 syncing with GPS for ~\$150 using a Raspberry Pi. And even with 300+ computers getting their time from it, it's hardly "taxed". I have yet to see the load for my Raspberry Pi, which is part of the NTP pool project, exhibit a load of anything more than 0.3, and that's with ~100 Kbps network traffic.

4. [Aaron Toponce](#) | November 27, 2013 at 2:39 pm | [Permalink](#)

Case in point:

```
$ ntpq -pn
      remote          refid      st t when poll reach    delay    offset    jitter
=====
-174.127.117.113 216.218.254.202  2 u  173  256  377   53.934   6.253   1.208
-209.236.69.170  231.146.174.254  3 u  164  256  377   53.804  15.409   6.325
-66.241.101.63   132.163.4.103   2 u  172  256  377   62.746   4.980  44.817
+199.104.120.74  85.29.25.171   2 u  183  256  377   64.776  -0.644   1.587
-63.211.239.58   128.138.140.44  2 u   61  256  377   62.611   7.154   4.624
*166.70.136.41   .GPS.           1 u  187  256  377   65.279  -0.343   0.797
+198.60.22.240   .GPS.           1 u   90  256  377   64.965  -0.444  14.843
+166.70.136.35   166.70.136.41  2 u   94  256  377   64.926  -0.288  17.398
-204.2.134.163   129.250.35.250  3 u   58  256  377   68.616   1.991  11.210
-178.18.16.124   204.235.61.9   3 u   95  256  377   70.709   3.189  23.020
```

Notice that the stratum 3 servers have high offsets, and high jitters. Even some of the stratum 2 servers are exhibiting high offsets with high jitters. This is fairly common.

{ 2 } Trackbacks

1. [Network Time Protocol | Share our secret](#) | December 12, 2013 at 1:18 pm | [Permalink](#)

[...] Real Life NTP (pthree.org) [...]

2. [Links 2 | SandzwerG.de](#) | March 22, 2014 at 4:09 pm | [Permalink](#)

[...] Real Life NTP Beitrag über die Funktionsweise und Einrichtung von NTP Servern und Clienten. [...]