NTP TIME

(WHAT TIME IS IT REALLY AND WHY DOES IT MATTER?)

TLARC MEETING 12/18/19 BY WK9M

(REVISED: JAN 1, 2020)

WHAT WE'LL COVER TODAY

- Keeping track of time in the past (WWV)
- How computers need more accurate time
- Built-in software vs. 3rd party software
- Home made and commercial NTP servers
- Meinberg's NTPd easy configuration
- (Optional) NTP in-depth server operation and custom configuration
- (Optional Demo) Meinberg's NTPd installation and configuration on a new PC

HOW WE GET THE TIME NOW

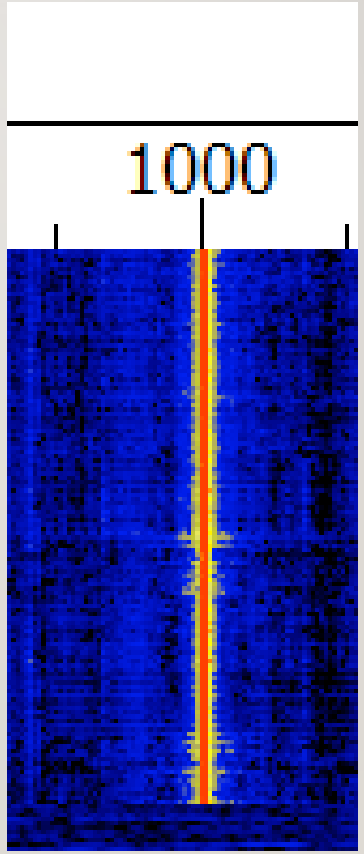
- WWV is the standard that most of us have grown up with; per Wikipedia they began broadcasts in 1945. WWV transmits on 2.5, 5, 10, 15, 20, and 25MHz while WWVB (used to set those inexpensive 'atomic clocks' you hang on the wall) transmits at only 60kHz.
- Both WWV and WWVB are detailed in full on Wikipedia and I'd invite you to visit these web pages when you get home if you'd like more information. I could spend the rest of this presentation on in-depth details but that's not the scope of this presentation today.
- There was talk about not funding WWV transmissions but that idea did not go through. For us hams, WWV is more than setting your clock.
- Let's have a quick listen to a WWV recording as a refresher so you can pick out the ticks, silence, and exact second marks.

OK SO THAT'S NOT REALLY WWV…

- Here's an actual WWV recording off the internet, courtesy of:
  - https://swling.com/blog/2014/08/listening-to-wwv-at-the-source-fort-collins-colorado-usa/
- There are other sources of time besides WWV such as WWVH (Hawaii), CHU (Canada), and DFC77 (Germany).
- All of these are variants are also detailed on Wikipedia.

FREQUENCY STANDARD OF WWV

- As a side note, WWV and similar AM broadcasts are useful for more than just time. Since they transmit a carrier, that carrier can be used to check the frequency accuracy of your receiver. i.e. this was captured off my radio using WSJT-X. I am at 9.999MHz USB on the VFO and the carrier is 1kHz higher. Looks good to me! (Would attenuate for actual test).
- WSTJ-X has detailed instructions on how to calibrate your RX using multiple frequencies…for another future topic perhaps.
- If your radio has an optional TCXO; get it! ☺

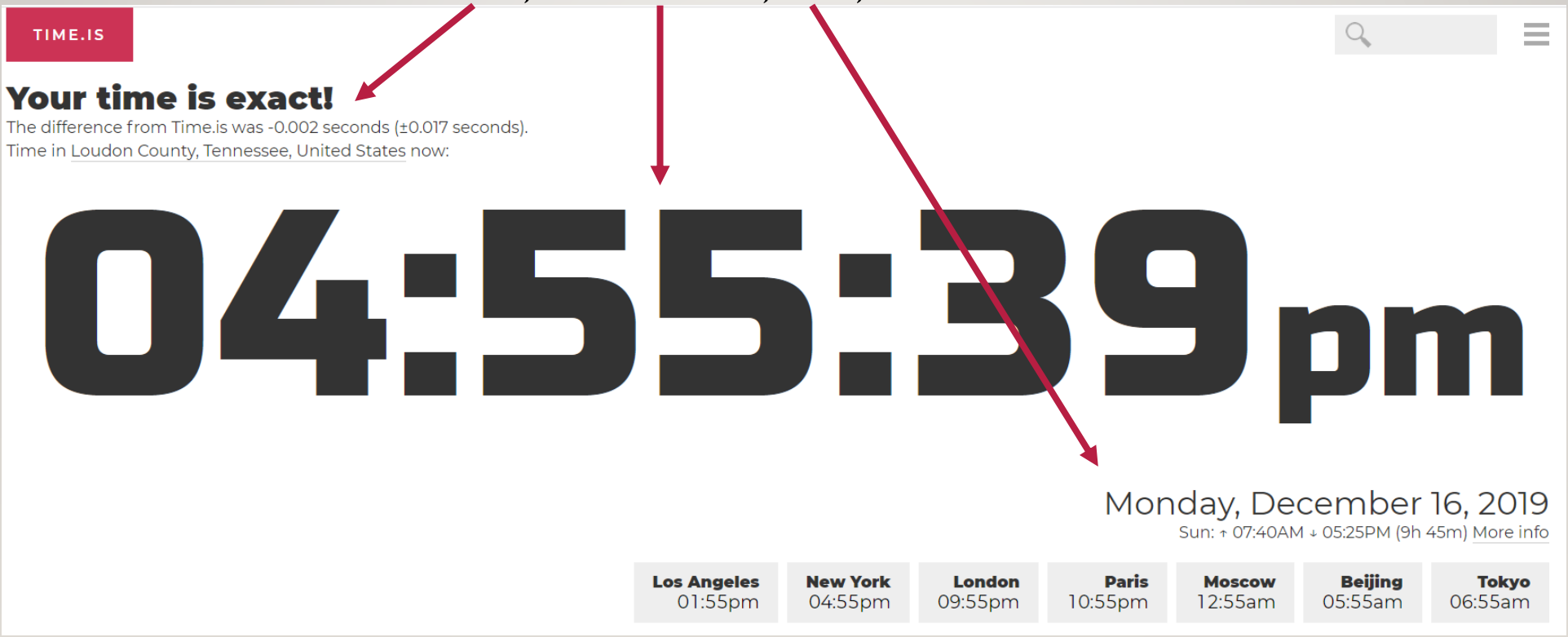


WHY DO COMPUTERS NEED ACCURATE TIME?

- The invention of computers and the need for accurate time becomes apparent fairly quickly for several services. Setting your PC by cell phone or watch isn't accurate for:
  - Surveillance—video cameras that could record the time of a crime
  - Banking—the exact time of a transaction or stock trade is critical
  - Ham Radio—logs only need to be within 5 minutes to match but WSJT-X has to be within 2 seconds for FT8.
- Quick check: How do you know how accurate your PC or device is right now? A great tip (web site) to check from QST is: https://time.is as shown on the following page.

SCREENSHOT OF TIME.IS REVEALS

- Difference from NTP time, current time, date, and other time zones.

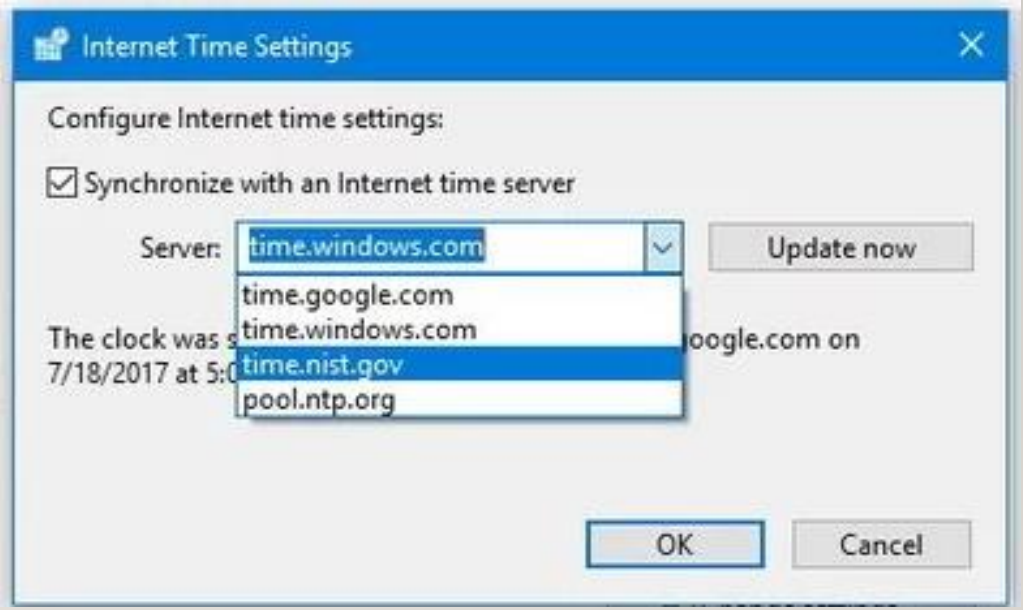


NTP TIME: WHAT IS IT?

- NTP stands for **N**etwork **T**ime **P**rotocol.
- NTP was invented a little before 1985 per the Wikipedia article; again you can read all of the background details on it again if you like. This is how computers and devices sync their time off the Internet now and I believe why WWV funding was iffy. Your computer doesn't have a radio built in it to receive WWVB but it has an Ethernet port (or WiFi).
- NTP devices set their time as the client; they connect to the NTP server to request the current time on port 123. This is similar to how you will connect to a web site on port 80, or the encrypted port on port 81.
- The NTP server feature is critical. Your WWVB clock does not have one; nor do most GPS clocks that direct decode the data.

HAVE A WINDOWS PC? BUILT-IN, YOUR TIME IS OFF

- Windows PCs by default use time.windows.com in order to fetch the time off the Internet. The problem here is that Microsoft's server has occasionally been wrong and even if you change it, that time only corrects itself once every 7 days. Weekly is way too infrequently to hold to 2 seconds (FT8). Local servers sync every hour instead of once every 168 hours…hmm.



DO IT YOURSELF NTP SERVER

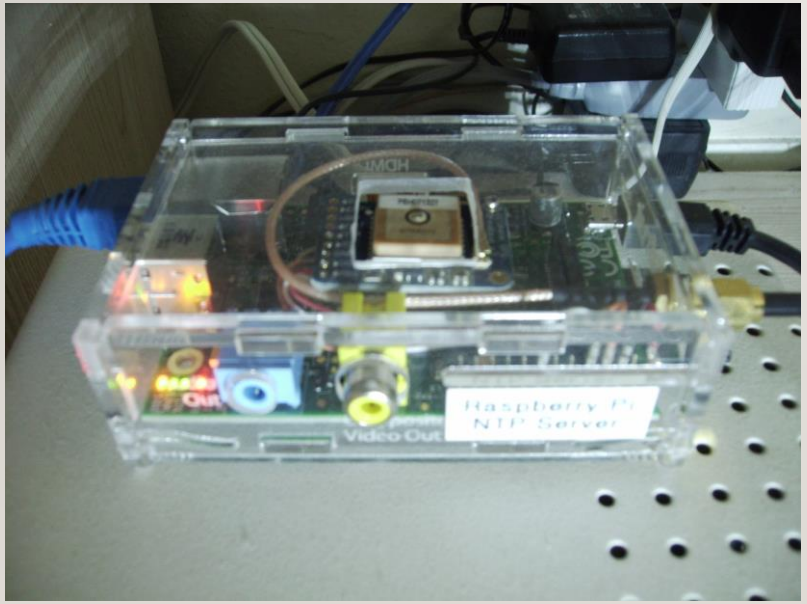
- A few years ago, with my spare time I decided that I wanted more accurate time without using the Internet and decided to build my own NTP server based off a Raspberry Pi.
- You can buy or build your own server that receives GPS time, and then uses a 1 PPS (Pulse Per Second) to sync the time exact. That is the difference between a GPS time and an exact GPS time. The data stream gets you close while the 1PPS will get you within 14ns per Wikipedia again. We don't need 14 billionth of a second accuracy, home built servers are easily under $100 while the commercial units start around $350 and rackmount units go into the thousands…but they all do the same thing.

DO I REALLY NEED MY OWN SERVER TO BE REASONABLY ACCURATE?

- Not really. There are two programs available online (Dimension4 and Meinberg's ntpd).
  - I have used Dimension4 in the past as it lets you specify multiple servers from the public ntp pool. A good server to use is us.pool.ntp.org
  - You can specify multiple NTP servers in the USA such as 0.us.pool.ntp.org, up to 3.etc
  - I've had problems with Dimension4 in the past dropping out servers and going out of sync.
  - Meinberg makes some of the expensive commercial units but they also offer a free NTPd client program that is a port of the Linux time server NTPd. Linux NTPd is more accurate because it not only accepts multiple servers but also marks bad ones and computes the time drift. Drift is the amount of time your device/PC differs and this value is constantly adjusted.
  - Dimension4 has a pretty user interface; Meinberg's requires you to edit a minor config file.
    - https://www.meinbergglobal.com/english/sw/ntp.htm

BACK TO THE HOME BUILT NTP SERVER IDEA

- I wanted something that didn't depend on the Internet connection, and I could request the time as much as I wanted to for multiple devices. Here's my RP server based off the instructions here: https://open.konspyre.org/2012-10-18-raspberry-pi-time-server/



This server was based off a RP Model B (2012) and it was a bit involved but a fun project to customize. The case is store bought with a hole in it but I had soldered and wired up the GPS RX to header pins myself and you have to recompile the kernel to add the 1PPS feature. Later versions of NTP outdated this model but it still works for demonstration purposes (and as a backup). To save time I'm not going over the instructions here. ☺

COMMERCIAL UNITS

- I won't show Symmetricom and Cisco rackmount units that cost 4-figures here. But there is another model from a company (Leo Bodnar electronics) out of the UK that works quite well. I bought one of these myself when I thought the Raspberry Pi server quit working—it was actually a bad remote GPS antenna (this GPS RX works with both internal and external automatically).
  - https://store.uputronics.com/index.php?route=product/product&path=60_70&product_id=92
- Price (after pound to dollar conversion) with shipping is around $352 now.
- This unit doesn't include an external GPS antenna, RG58 BNC jumper for the external 10Mhz/1PPS source, or a USB power supply at that price (will with with PoE if you have it). But, any compatible antennas or jumpers will work as well.

LEO BODNAR'S NTP SERVER

- Show time, clients, GPS stats, network config info, load, etc.
- Can output 10MHz, 1PPS, etc.
- Firmware updatable; not Win/Linux.
- Fast GPS sync: 30s or less cold start
- Handles 100K requests/s. Overkill?
- This model has an OLED display
  - OLED can screen burn w/ static images

CONFIGURING MEINBERG'S NTPD

- In order to show briefly that the Meinberg software isn't that much work to setup, here is the ntpd.conf file and the one line you need to look for when installing.
  - # Use specific NTP servers
    server <default> iburst <optional other stuff>
  - Comment out the above and change to the pool line(s) to show instead
    pool 0.us.pool.ntp.org iburst minpoll 12
    pool 1.us.pool.ntp.org iburst minpoll 12
    pool 2.us.pool.ntp.org iburst minpoll 12
    pool 3.us.pool.ntp.org iburst minpoll 12
  - UNLESS you travel to other countries with the device/PC, then use the generic pool.ntp.org and/or the numbers 0-3 as needed.

DETAILED NTP SERVER OPERATION

- At this point, we've reached the end of the generic part of the presentation that most people would like background info on.
- If you have other plans you can easily leave now and not miss much. There's 6 pages left.
- If you want to see some demos and configuration on what actually happens in the background I'll proceed with NTP server operation.
- <PAUSE A BIT HERE; bathroom break anyone?>

NTP SERVER OPERATION

- The NTP server itself has a configuration file that is unique to that server's application. You can share the NTP time online, restrict it to just local computers, use other servers as part of your server, etc. i.e. the RP (prior to the 4B) uses Ethernet over USB and compensates for that by fudging the time .496ms so it starts out more accurate.
- The master clock goes by Stratum number. A stratum 0 source is WWV for example, or a GPS satellite with a cesium standard. Stratum 1 is another 'server' that is based off a stratum 1 clock; i.e. your own PC on a stratum 0 source would be stratum 1. Stratum 2 is a server that runs off a stratum 1 source, and so on. These numbers increment until at some point you no longer trust the accuracy; stratum 16 is a free-running server that has a time to give but it's free-running off a crystal source.
- To 'be nice', don't use stratum 0 sources by default unless you have a specific need.

NTPD SERVER AND NTP CLIENT STATUS

- The command to check the status on the server is ntpq –p.
- ‘remote’ denotes the type of device; in this case it’s GPS based along with the refid. The little ‘o’ means that 1PPS is in use.
- st means my NTP server is a stratum 0 device; same as WWV. t is the souce; ‘l’ for local. when is the last request (in seconds); poll the time (in s), reach is a flag for good data (in octal 377 is 8-1s), delay (in ms), offset is difference between server and client (ms), jitter is used to indicate difference between two samples (clock stability or RMS of offsets).

```
pi@raspberrypi /etc $ ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
oGPS_NMEA(0)     .GPS.           0 l    5    8  377    0.000    0.000   0.004
pi@raspberrypi /etc $
```

ON THE CLIENT (PC) SIDE

- The PC looks pretty similar to what the server shows but you'll notice differences.
- The remote shows the name or IP address of the server now. * shows the current time source and you'll notice that 't' is unicast (off my LAN) while poll is set to 1024 seconds. With a local server I set maxpoll instead of minpoll to 12 or 2^12 seconds=4096 (68 minutes). On the internet servers if you don't limit it to an hour (maxpoll) you will send out a lot of request; on a local server I don't care. The system decided on 1024 seconds after a few hours or around 17 minutes. Over time this can increase. Still, under 7.5ms.

```
Checking current status of NTP service with ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*LeoNTP          .GPS.            1 u  369 1024  377    0.409   -6.048   0.980
(Auto-Refresh every 10s  --- CTRL+C to Cancel)
```

JUST ONE NTP SERVER?

- Wait! I said you need at least three NTP servers. That is correct when you're using the Internet as a time source. One will become the reference and the other two supplemental timings. If one is off from the other two, that one gets tossed out (marked as bad) and isn't used until a reboot or restart of the service. Using different numbers in front of pool.ntp.org forces unique servers, so you never get the same one twice.
- When running a local server however as a stratum 0 source that time is taken as golden because it is separate and isolated from the Internet.
- If you are running a high security installation however, three local servers would be safer from hacking. NIST also has an encrypted NTP server. Leo Bodnar claims that their NTP server doesn't use Linux or Windows so it can't be hacked…sure OK it is. ☺

NTPQ –P DETAILED

- If you want to know what all of the options and symbols mean, the best place I've found is this explanation online. I checked my memory with this web page below and added a few tips http://tech.kulish.com/2007/10/30/ntp-ntpq-output-explained/
- There is also an NTP configuration 'cheat sheet' at the same Meinberg download page.
- Here's a few lines from the above showing the more typical installation and some bad ones (-)

```
remote          refid      st t when poll reach   delay   offset  jitter
-navobs1.oar.net .USNO. 1 u 958 1024 377 89.425 -6.073 0.695
*navobs1.gatech. .GPS. 1 u 183 1024 375 82.102 1.639 0.281
-NAVOBS1.MIT.EDU .PSC. 1 u 895 1024 377 90.912 -0.207 0.368
+navobs1.wustl.e .GPS. 1 u 48 1024 377 76.890 1.093 0.525
```

NTP SERVER AND CLIENT SETUP AND DEMO

- This is finally the end of the presentation; thanks for attending and hope you picked up some good info that you can use.
- If you stick around a bit and would like to see a demo pending available time, I'll install the Meinberg NTP client from scratch on this PC, configure ntpd.conf, and connect to the Raspberry Pi NTP server to see how accurate we can get it. (In the time we've spent at the presentation, the Raspberry Pi server should have a pretty accurate drift file).
- See you next time!
- Randy, WK9M