Time Distribution Aspects

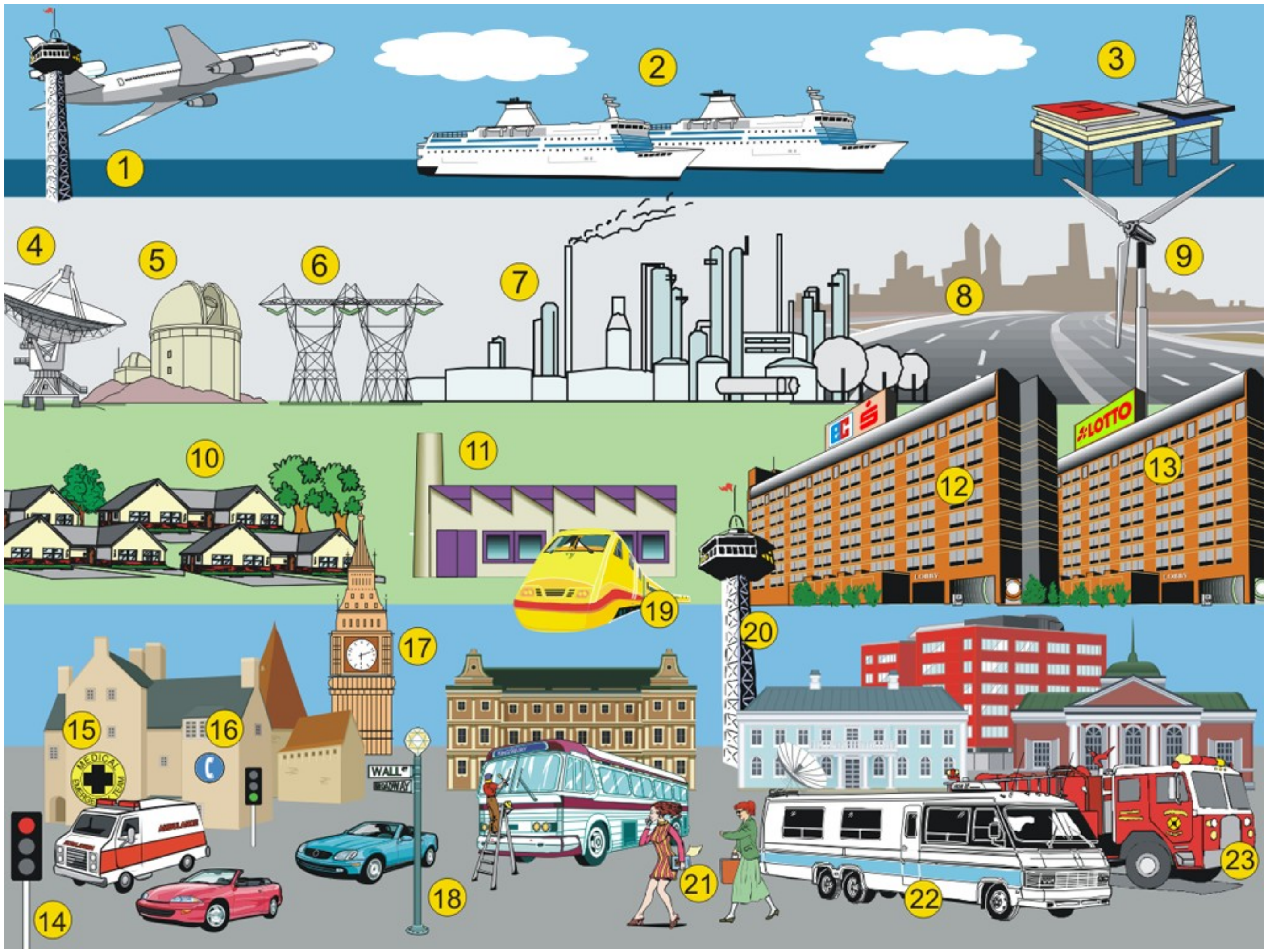
Martin Burnicki
martin.burnicki@meinberg.de

Meinberg Funkuhren GmbH & Co. KG
Bad Pyrmont, Germany
https://www.meinbergglobal.com

Agenda

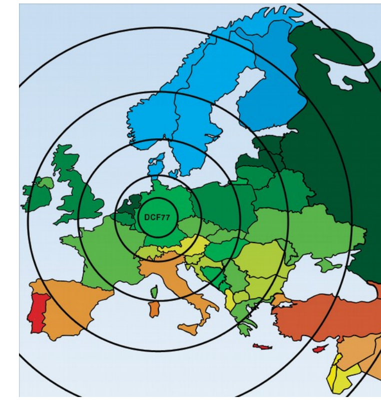
- Some Basics
- Reference Time Sources
- Ways to Distribute Time
- Network Time Protocols
- Hardware Signals
- Accuracy considerations

Who Needs Time Synchronization?



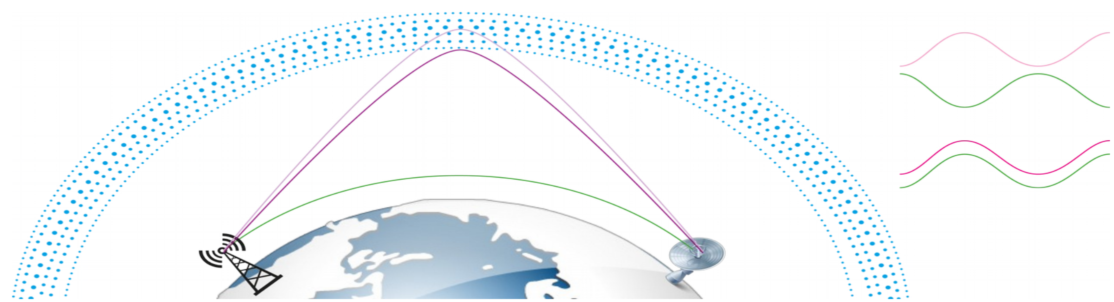
1. Air Traffic Control
2. Research Vessels
3. Oil Production
4. Satellite Communication
5. Observatories
6. Power Substations
7. Power Plants
8. Toll Charging Systems
9. Wind Energy Plants
10. Public Infrastructure
11. Production Flow
12. Banks, Cash Terminals, Stock Exchange, Data Centers
13. Lottery
14. Traffic Management
15. Operation Coordination
16. Event Management
17. Wall Clocks
18. Lighting Control
19. Railway Time Table
20. Radio Broadcasting
21. Mobile Communication Call Data Records
22. Outside Broadcast Van
23. Emergency

Reference Time Sources (1)



Long Wave Receivers

- DCF-77, MSF, WWVB, JJY
- Single transmitter, delay compensation not easy
- Interference ground/sky wave may cause signal cancellation
- Varying signal propagation delay, low bandwidth
- Milliseconds accuracy only

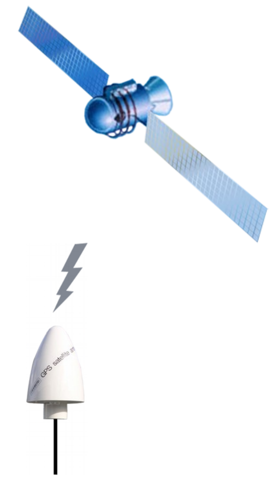


Telephone Modem Services

- ACTS operated by NIST, PTB
- No antenna required, but telephone line
- Eventually phone costs per call
- Milliseconds accuracy only

Reference Time Sources (2)

Satellite Systems



- GPS, GLONASS, Galileo, Beidou
- Multiple satellites per system tracked simultaneously
- Propagation delay can be measured and compensated
- Tens of nanosecond accuracy internally
- Even higher accuracy with specific methods

NTP/PTP Time Servers

- Usually get reference time from one of the other sources
- Accuracy depends on accuracy of upstream source

Basic Question:

→ **How to get the accurate time into the system/application?**

Time Synchronization Aspects

Where do I get the time from? At which accuracy and precision?

- Long wave radio clock (DCF77, WWVB, MSF, JJY, ...)
- Telephone services (ACTS by NIST, PTB)
- GNSS satellite receiver (GPS, GLONASS, Galileo, Beidou, ...)
- Time server / network protocols (NTP, PTP, “White Rabbit”)

Which ways to get the time?

- PCI card/USB: Can always read the current time immediately
- Serial/USB: Wait for time string. When sent? Transmission delay?
  - ➔ Similar to strokes of a church bell
- Hardware Pulses: Acquire system timestamp when on-time slope detected.
  - ➔ How is slope detected? Latencies?
- Network / ACTS: Send query, wait for reply, try to determine packet delays

➔Need to compensate network delay and other transmission delays

Time Client Considerations

Resolution of the local system time

- Milliseconds, microseconds, or nanoseconds

Possible ways to adjust the local time

- Change timer increments/reloads
- Control oscillator frequency

Stability of the on-board system clock

- Quartz frequency offset, variations with temperature, PC virtualization

Time synchronization software

- Which resolution is supported?
- Is transmission delay compensated?

➔**How is client time adjusted?** Set periodically? Smoothly?

➔**Very important: Handling of Leap Seconds by server and client**

System Time Resolution

On some operating systems limited to timer tick

- Windows XP / Server 2003: about 16 ms
- Windows Vista / Server 2008: 1 ms
- ➔ Reading system time yields same time during timer tick!
- Since Windows 8 / Server 2012: Alternate API call with sub-microsecond resolution

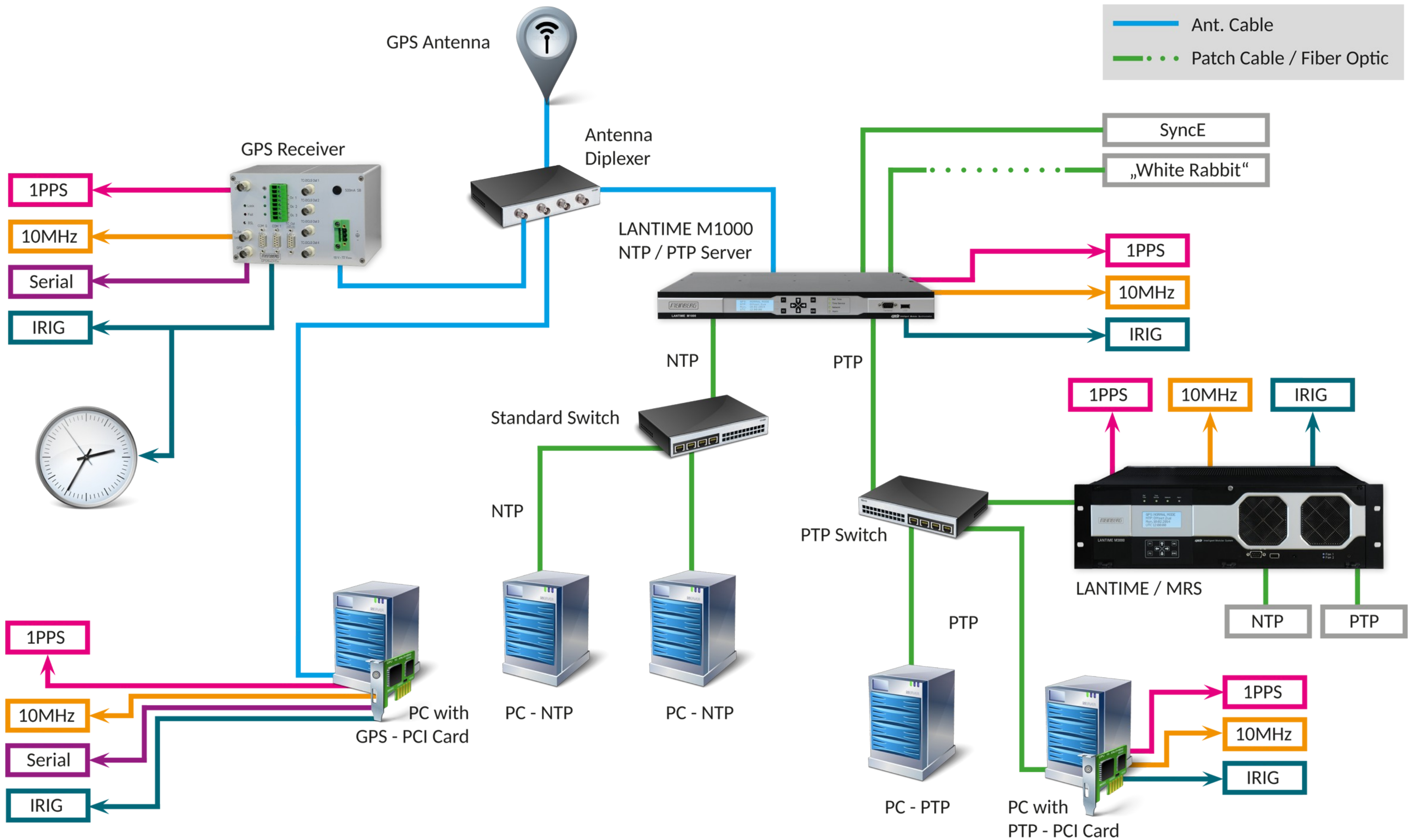
Other operating systems provide better resolution

- Linux / Unix: microseconds or even nanoseconds
- ➔ Reading system time yields always different timestamps

➔Nanosecond resolution does not necessarily mean nanosecond accuracy and precision, but high resolution is a precondition

➔In any case limited accuracy with virtual machines

Time Distribution (1)



Time Distribution (2)

GPS controlled NTP/PTP time server

- Provides clients with very accurate time
- Single antenna cable required
- NTP with good accuracy for “normal” servers
- PTP with highest accuracy for special requirements
- PTP PCI cards get high accuracy into a server
- PTP-aware switches required for high accuracy
- Standard patch cables instead of special antenna cables
- Optionally additional hardware output signals (1 PPS, IRIG, ...)

Time Distribution (3)

GPS receiver card in each server, multiple GPS antennae

- High accuracy
- High cabling efforts for antennae
- Independent operation

GPS card in each server, single GPS antenna with diplexer(s)

- High accuracy
- Only single cable to antenna, but special antenna cables to each GPS card
- Dependency on single antenna

Hardware signals from time server or PCI card

- Accuracy depends on signal type (IRIG, 1 PPS, ...)
- Well-suited for hardware-based applications
- Special cabling depending on signal type
- Usually care about polarity and signal level

Network Time Synchronization Protocols

Network Time Protocol (NTP)

- Invented later in the 1980s, 0.2 ns resolution → supports high accuracy
- Current protocol version is v4, compatible with older versions
- Standard protocol for time synchronization in Unix/Linux/*BSD, and Windows
- Reference implementation available as free software

Precision Time Protocol (PTP/IEEE1588)

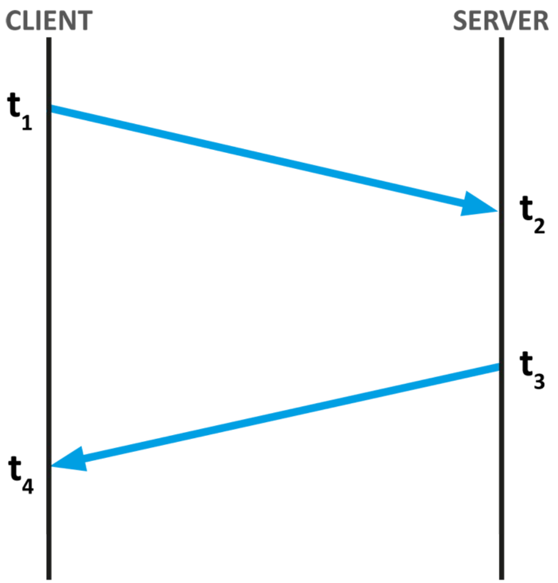
- v1 from 2002, v2 from 2008, v2 is not compatible with v1
- Nanosecond resolution, eventually yields some nanoseconds accuracy under specific conditions (e.g. hardware timestamping on **every** network node)
- Open source implementations available

New "White Rabbit" Project

- Sub-nanosecond accuracy over Ethernet
- PTP over Synchronous Ethernet (SyncE)
- Both hardware and software available as Open Source

Example: NTP protocol

t1: Client sends request packet to server
t2: Server receives request packet from client
t3: Server sends reply packet to client
t4: Client receives reply packet from server



→ Four timestamps from one packet exchange

→ Timestamps from server are server time

→ What's the offset between server time and client time?

→ How long did the request and reply packet travel on the network?

→ High accuracy if the network delays are the same in both directions

Network Packet Latencies (1)

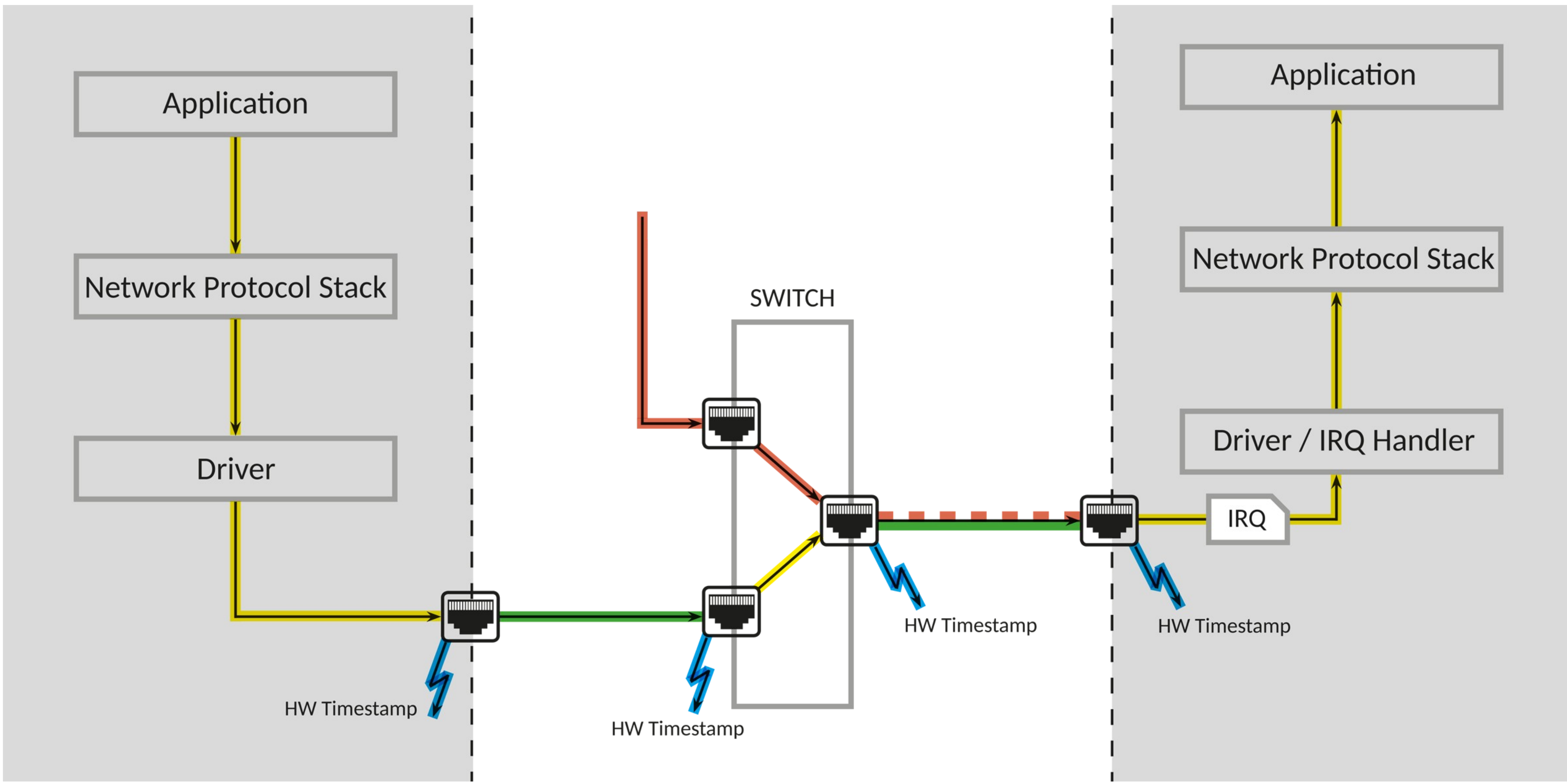
- Processing time of packets on server and clients
- IRQ latencies of high performance NICs
    - ➔ Interrupt coalescence, IRQ priority
- Delays in routers and switches
    - ➔ Packets can be queued, or not
    - ➔ Different link speeds on ingoing/outgoing port

➔ Hardware Timestamping can reduce variable network delays

➔ Required on every network node to get highest accuracy

➔ Use PTP with special PTP-aware switches

Network Packet Latencies (2)



Network Time Client Requirements

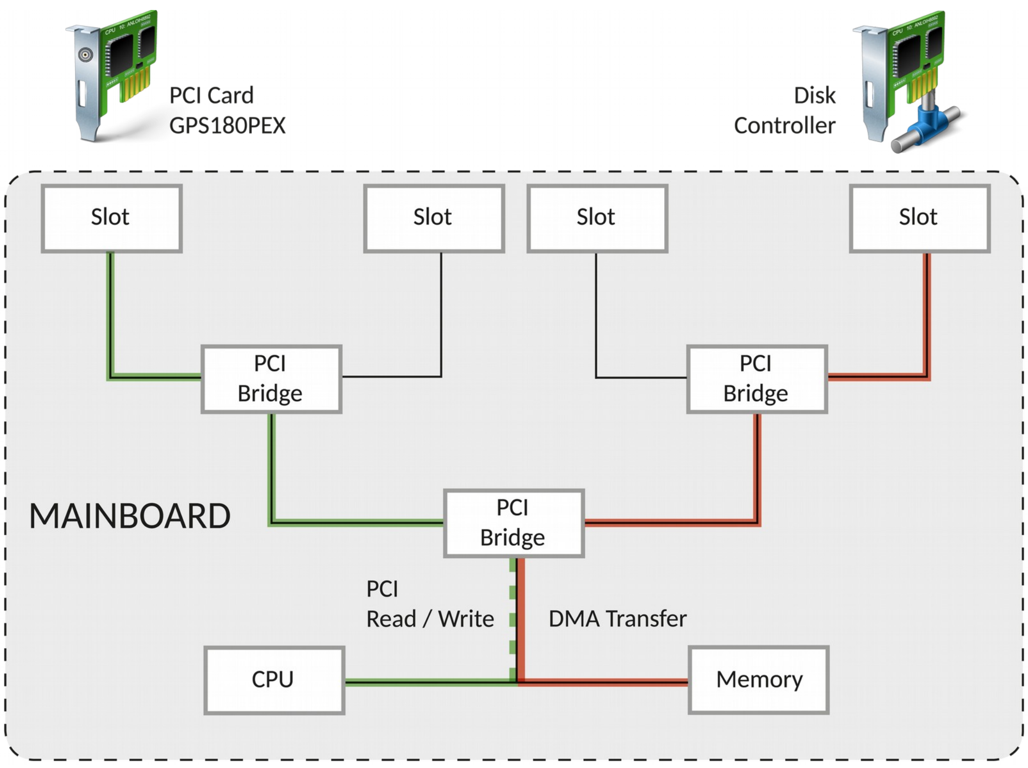
- Network delays are not constant → **filtering** required on the ***client***, or hardware timestamping to eliminate jitter.

- Achievable accuracy does not only depend on the accuracy of the server …

- It depends strongly on the **client hardware and software implementation**. Only client can compensate delays.

➔ When talking about NTP or PTP distinguish between **protocol** and **implementation**.

PCI Access Latencies



- PCI busses are similar to local networks
- PCI bridges correspond to network switches
- Bus arbitration can cause delays like queuing in switches
- If the bus is busy then other access is delayed
- PCIe access time few microseconds, depending on chipset on mainboard
- Software execution can be interrupted

Hardware Signals (1)

1 PPS slopes

- No absolute time, only second boundaries

10 MHz frequency

- Sine wave or square wave
- No time information, no second boundaries
- Can avoid client clock drift

Serial time strings

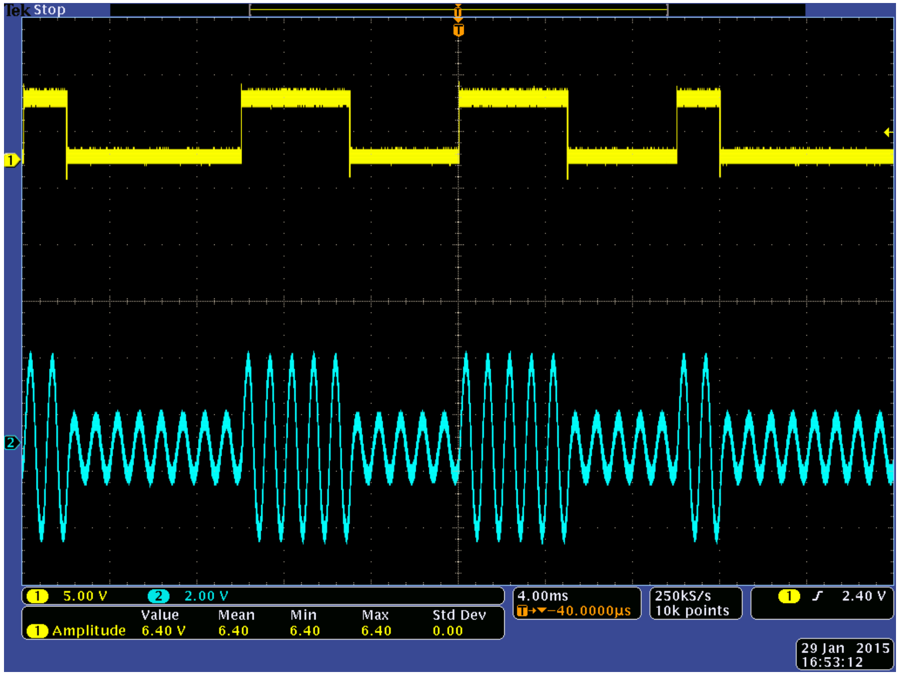
- String format, NMEA or different, vendor-specific
- Which information is included?
    - → Date, time, DST/Leap second flags, UTC offset, ...
- Fixed timing relationship to second boundaries, or not?
    - → Some 3rd party GPS receivers don't have it --> error prone

Hardware Signals (2)

IRIG/IEEE time codes

- Unmodulated and modulated codes/signals have been specified
  - Modulated codes easy to handle, can be distributed from one output to several inputs, and even be recorded e.g. on magnetic tape. No need to care about polarity.
  - Unmodulated codes to be distributed over same physical ways as 1 PPS, but easier to yield higher accuracy than modulated. Polarity, level!
- Transported information depends on code format
- Provides at least day-of-year and time
- Many commonly used codes don't provide year number, UTC offset, or leap second information
- IEEE 1344 / C37.118 codes provide 2 digit year number, UTC offset, DST and leap second flags, etc.
  - Unambiguous decoding
  - Compatible with popular IRIG-B002/B122

Unmodulated vs. Modulated Time Code



- Unmodulated codes:
  - Fast slopes, thus easy to yield accurate time
  - Usually more
- Modulated codes:
  - Zero crossing of sine wave to be detected accurately
  - Can be recorded on magnetic tape

Hardware Signal Distribution (1)

- 5V/TTL, RS-232
  - Asymmetric signals, susceptible to electric noise
  - Short distances only
  - Insulation requires specific circuitry

- RS-422/RS-485, current loop
  - Symmetric signals, good noise suppression
  - Longer distances possible
  - Insulation requires specific circuitry

- Fiber Optic
  - Not susceptible to electric noise
  - Very long distances possible
  - “Built-in” insulation

Hardware Signal Distribution (2)

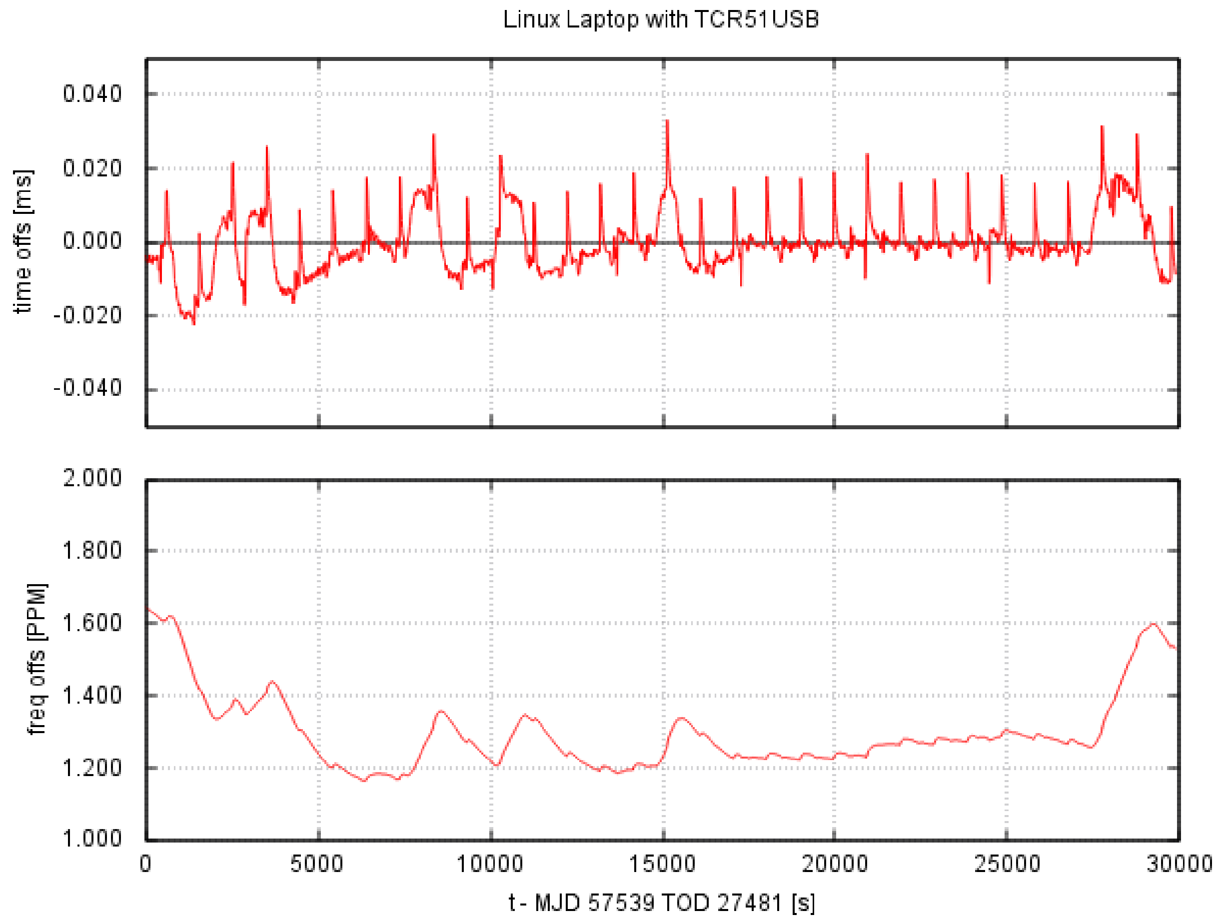
- PPS over USB (Serial-To-USB-converters) or even WIFI
  - → PPS slope is emulated by USB messages
  - → RF transmissison introduces jitter and delay
  - → Accuracy is reduced in any case

- Serial time strings
  - → String format, NMEA or different?
  - → Which physical transportation?
  - → Fixed timing relationship or not?

Things to keep in mind:

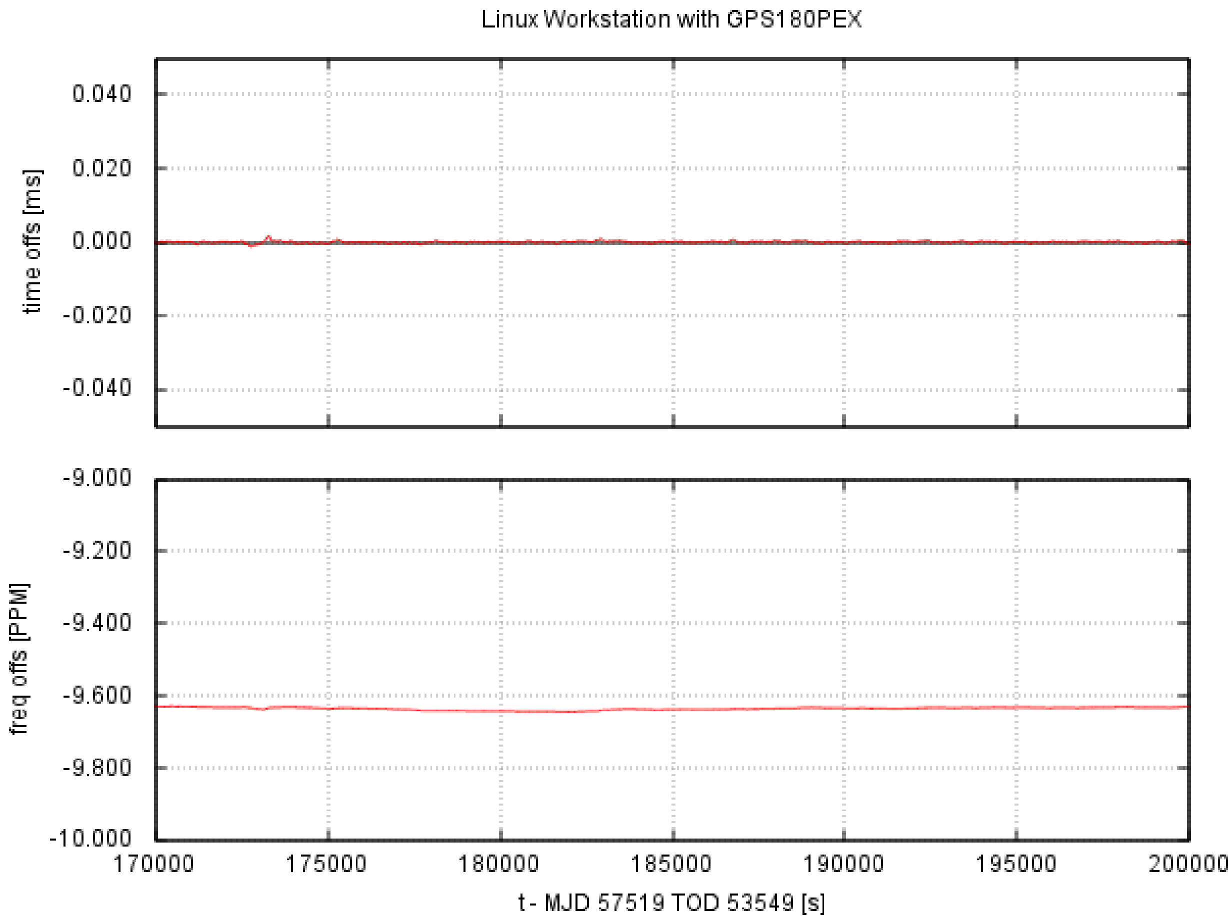
- → Type of transmitter and receiver have to match, even polarity
- → Insulation often required to avoid ground loops
- → Which is maximum supported distance?

Linux Laptop with TCR51USB



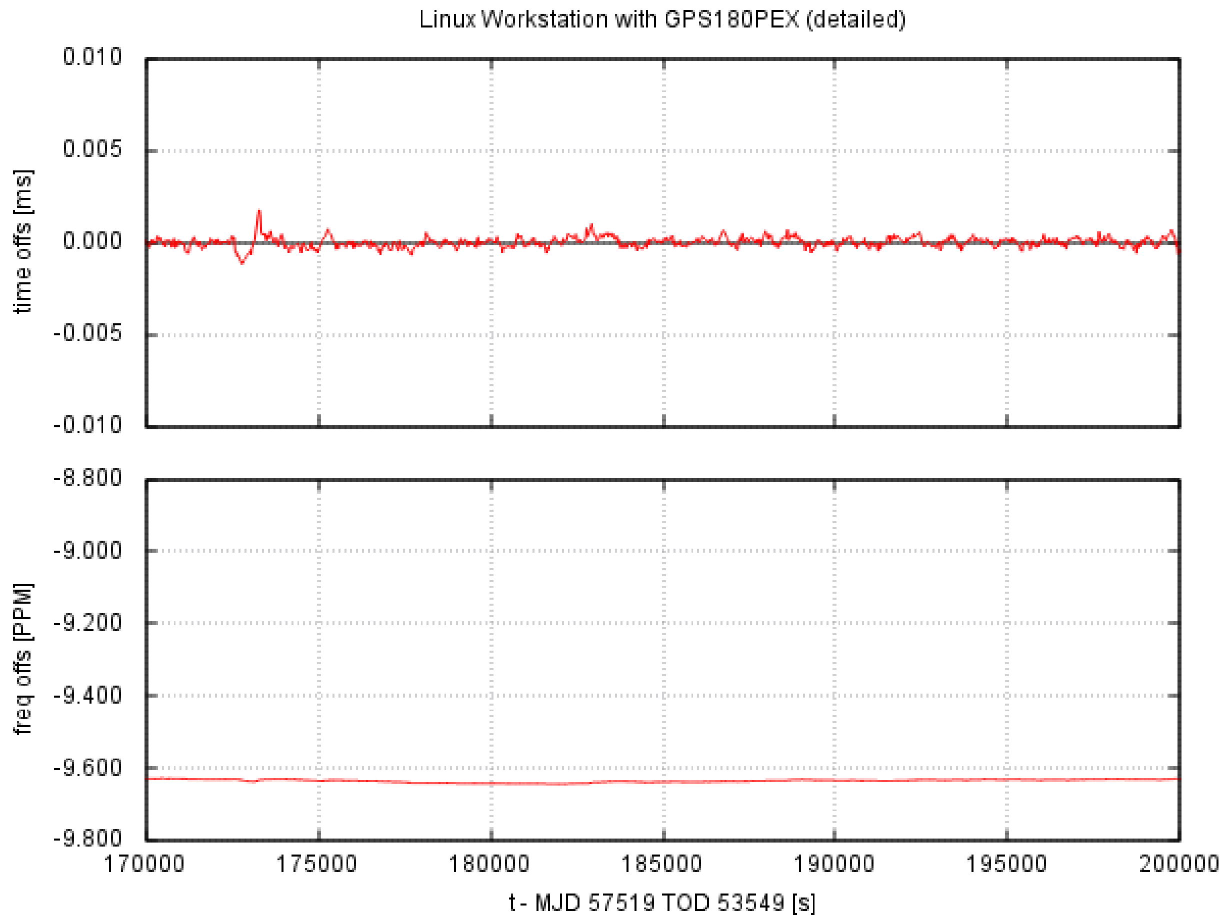
- Jitter caused by USB bus is flattened by ntpd's filtering
- Anyway results are pretty good under given conditions

Linux PC with GPS180PEX



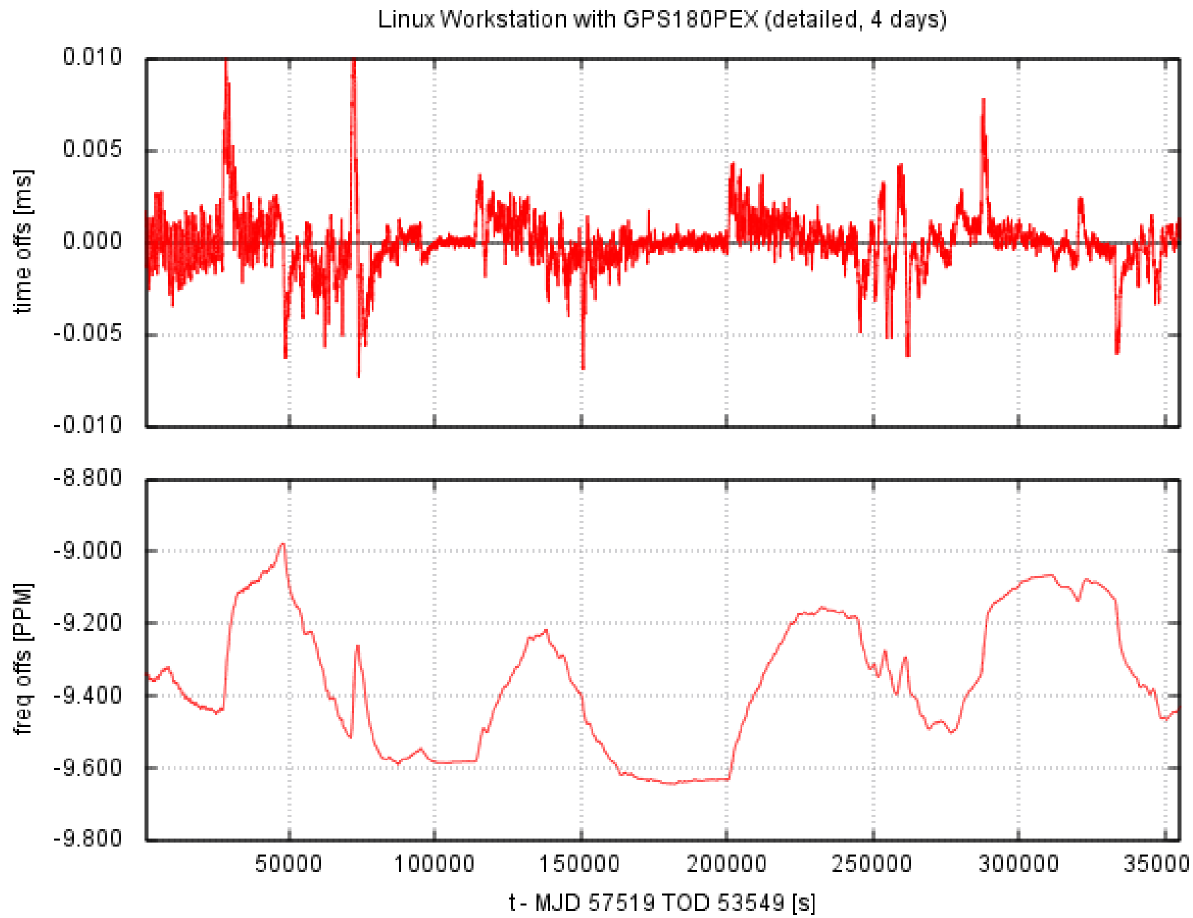
- Similar measurement with GPS PCI card in Linux PC
- Looks even better -> Look with increased scale

Linux PC with GPS180PEX, detailed



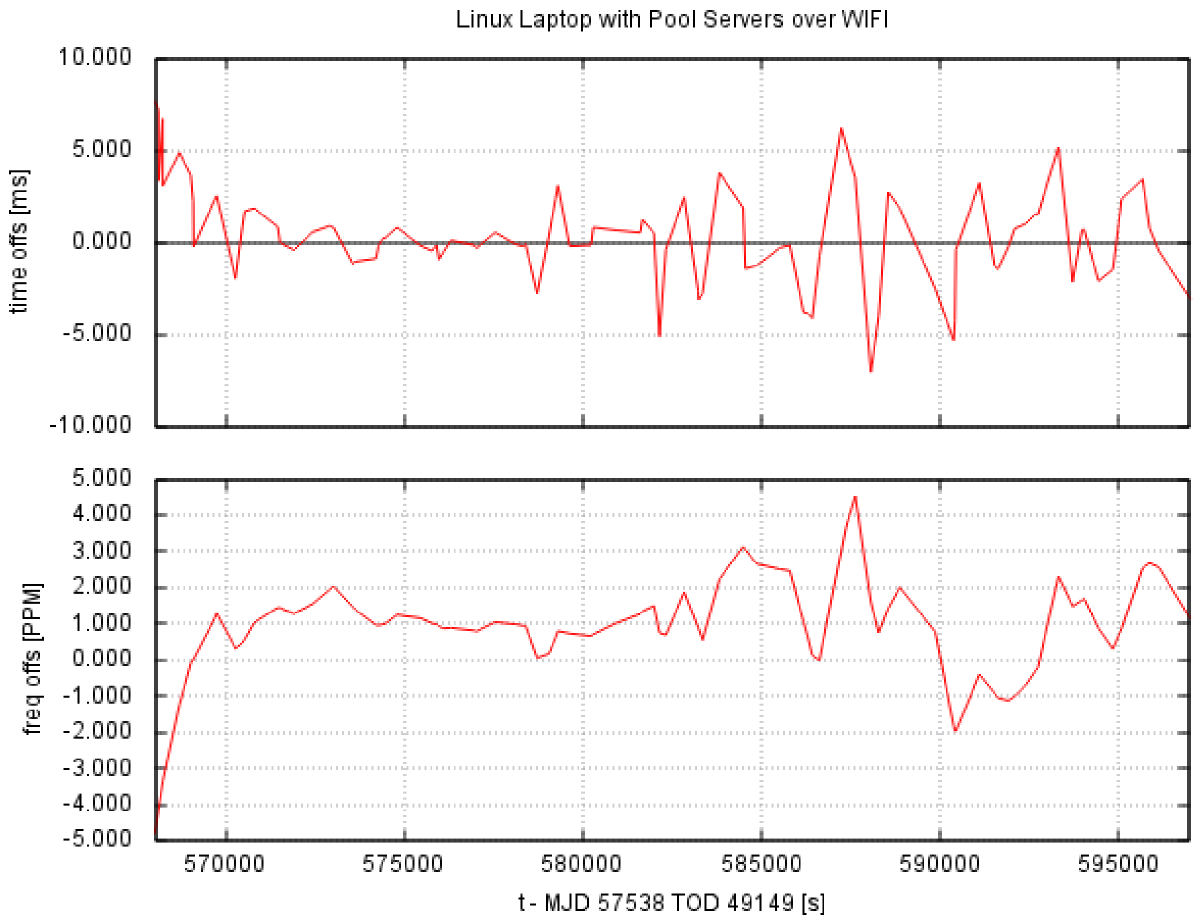
- Same measurement as before, but finer scale
- Frequency offset constant means temperature is constant

Linux PC with GPS180PEX, 4 days



- Measurement interval 4 days
- Frequency variations due to daily temperature changes
- As a result less time accuracy than for short term

Linux Laptop, Pool Servers over WIFI



- Time offset scale is now +/- 10 **milli**seconds, not **micro**seconds
- Accuracy still good under given conditions
- ➔ Can be better, but depends on requirements

Comparing Offsets Measured By ntpd

ntp.conf:

```
server 127.127.28.0 iburst minpoll 4 maxpoll 4
fudge  127.127.28.0 refid shm0
server 192.168.0.2 iburst minpoll 4 maxpoll 4 noselect
```

ntpq -p:

```
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*SHM(0)          .shm0.           0 l    6   16   377    0.000    0.001   0.002
 192.168.0.2     .MRS.            1 u    3   16   377    0.187    0.013   0.065
```

➔ **Microseconds difference between IRIG and NTP with direct connection**

Facts Affecting Time Accuracy

- Protocol or signal type
- Transmission
- Delay/latency compensation
- Protocol jitter
- Client Operating System
- Client Hardware
- Virtual Machines vs. Physical Machines

Conclusion

- Timekeeping accuracy depends on many facts
- Required accuracy depends on applications
- Higher accuracy requires higher effort, and thus solutions are usually more expensive
- Important to find the solution which best meets the requirements of your application

- Founded in 1979 by Werner and Günter Meinberg
- Initial product range: DCF77 longwave radio receivers for industrial applications (1980)
- First self-developed GPS time receiver in 1993
- Full-depth manufacturer: research, development, design, production, sales and support in one hand
- In-House production of 90% of the mechanical (chassis/housing) and electronic (modules, integration) components
- 90 employees (25 R&D), one central campus in Bad Pyrmont
- approx. 70 km southwest of Hannover, Northern Germany

Thanks!

Thanks for your attention!

Any Questions?

Also don't hesitate to contact Martin Burnicki, or Mark Street of JTime!, Meinberg's distributor in the U.S.