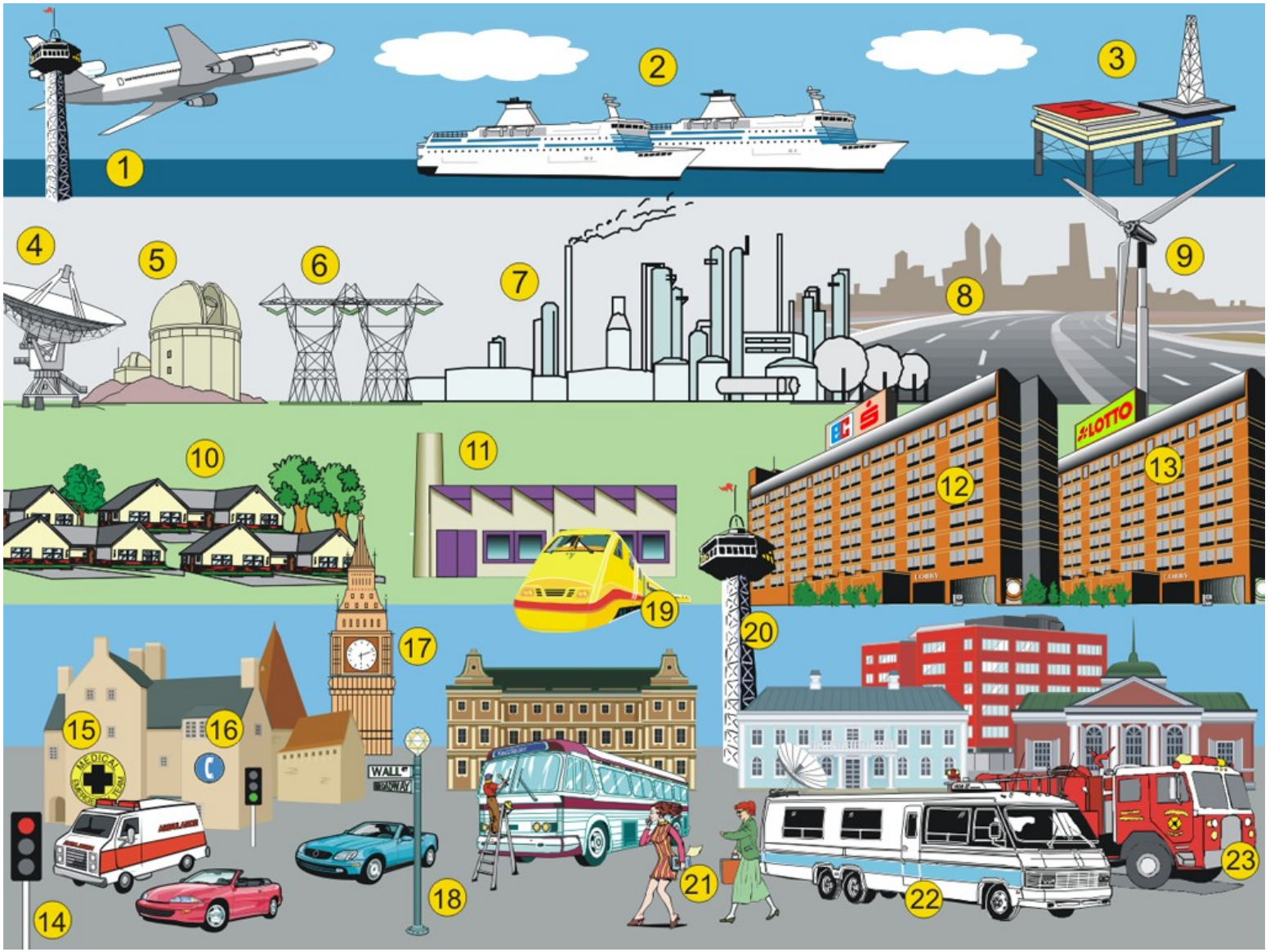
Computer Time Synchronization Concepts

by **Martin Burnicki**
Email: martin.burnicki@meinberg.de

Meinberg Funkuhren GmbH & Co. KG
Bad Pyrmont, Germany
http://www.meinberg.de

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

World Time vs. Local Time

The sun is expected to have highest elevation at noon

World has been divided into 24 time zones

Time zones usually differ by 1 hour

→ A few refgions have local times 15, 30, or 45 minutes off

Time zone borders often follow borders of countries

Many countries are in a single time zone

→ People often don't care about the time zone

Large countries span several time zones

→ People are used to account for different times in different zones

Time zones are derived from common world time (UTC)

UTC And Atomic Time Scales

Formerly a second was derived from earth rotation

Earth rotation speed changes, so the length of a second could vary, which is not useful for technical applications

Today length of second is determined by atomic clocks

Atomic clocks just count oscillations (no radioactivity!)

National standard institutes in many countries operate atomic clocks

World time UTC is derived from many atomic clocks world-wide

Leap seconds are inserted to compensate earth rotation changes

Local time zones are derived from UTC by applying an offset to UTC

Leap Seconds

Compensate earth rotation changes

Determined by IERS

Usually leap seconds are inserted, but in theory can also be deleted

Cause inconsistency of time

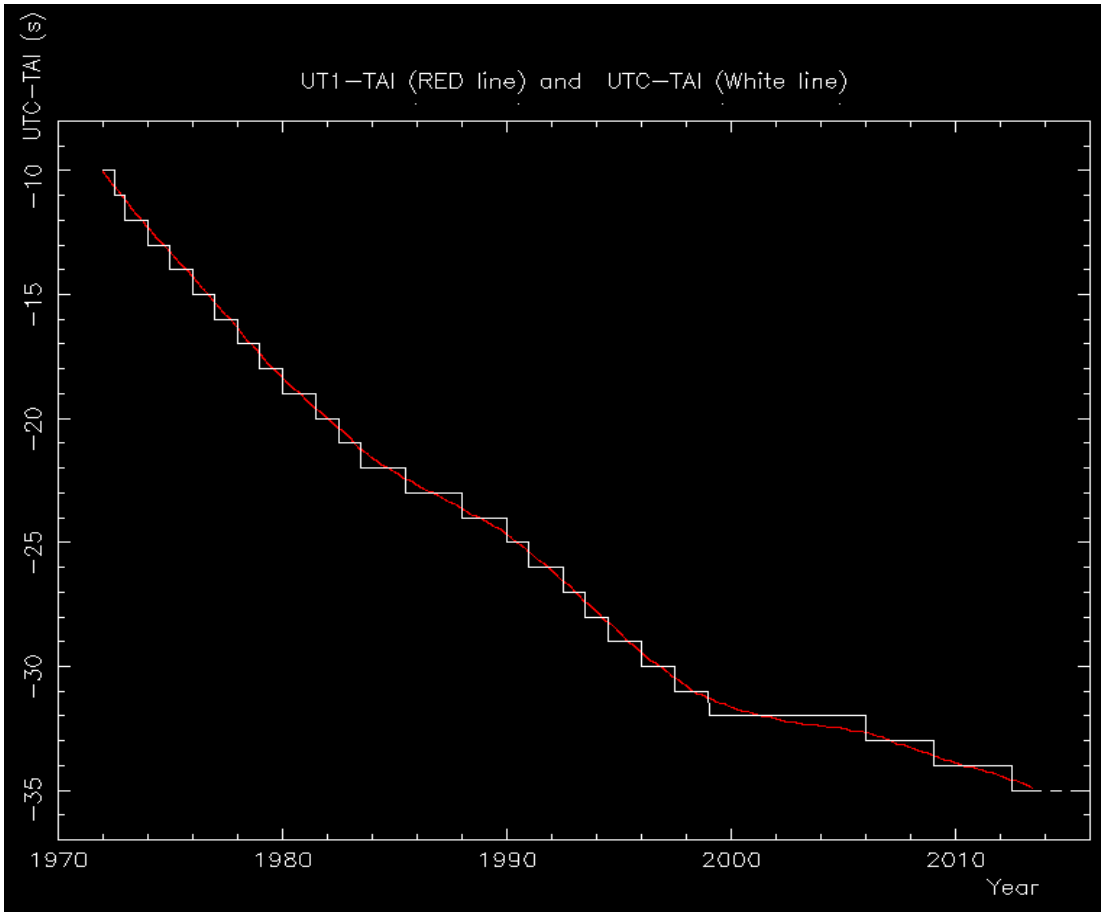
→ 23:59:60 same as 00:00:00

Causes different length of day

→ 1 second more than usual

Must be taken into account for billing systems, etc.

Must not be mixed up with leap years



Source: International Earth Rotation Service
http://hpiers.obspm.fr/eop-pc/earthor/utc/leapsecond.html

How Computers Keep Time

On-board RTC chip counts time while off

System time initialized from RTC at boot time

System clock counts in timer ticks

Timer ticks are derived from a cheap crystal oscillator

Oscillator frequency is usually more or less off its nominal frequency

Oscillator frequency changes with temperature

As a consequence of the frequency offset the system time drifts

Computer UTC vs. Local Time

Computer system time is usually kept as UTC

Converted to local time according to user preferences (configuration)

On Windows usually only a single time zone setting

On Unix/Linux even single processes can run with different time zone settings

Switching to/from daylight saving (DST) is done by the operating system

Time synchronization only adjusts UTC system time

If UTC time is correct then local times are also correct

Time zone parameters and DST are not job of time synchronization software

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!

Other operating systems provide better resolution

Linux / Unix: microseconds or even nanoseconds

Reading system time yields different time stamps

Microsecond resolution does not necessarily mean microsecond accuracy, but high resolution is a precondition for high accuracy.

E.g. it's a problem to have 1 ms accuracy with 16 ms resolution

How do I know which time it is?

MEINBERG

Do it the Italian way ;-)

[video]

Look at a clock which I have in sight

I can look at the clock whenever I want, as often as I want, and know the time immediately

→ Read fast time stamps from a PCI card

Listen when the church clock bell sounds

Whenever the bell sounds I know it's a full hour

After I have counted the strokes I know what time it was **at the beginning**

I don't know exactly what time it is **in between**.

→ Wait until serial time string sent automatically, e.g. on second changeover

Ask someone else who knows the time

The other person looks at its wrist watch and replies to my query earlier or later

→ Network time protocols (NTP, PTP)

→ Querying slow devices (USB)

Computer Time Synchronization Aspects

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

- **Radio clock**
- **Time server**

Which ways exist to get the time?

- **PCI card: Can get the current time always, immediately**
- **Serial: Wait for time string. When sent? Transmission delay?**
- **Network: Send query, wait for reply, compensate network delays**

Resolution of the local system time

- **Depends on operating system**
- **Windows: 16 ms (XP) or 1 ms (Vista and newer)**
- **Unix/Linux: 1 µs or even better**

Time synchronization software

- **Which resolution is supported?**
- **Is transmission delay compensated?**
- **How is system time adjusted? Set periodically? Smoothly?**

Network Time Synchronization Protocols

“time“ and “daytime“

From 1983, 1 s resolution, → limited accuracy

NetBIOS / NetBEUI

Proprietary Windows, used by early Windows versions only

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, and Windows

Reference implementation available as free software

Precision Time Protocol (PTP/IEEE1588)

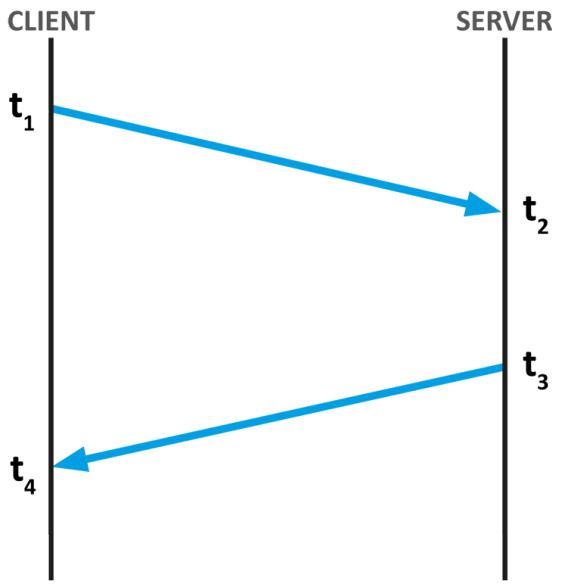
Can yield nanosecond accuracy under certain conditions

V1 from 2002, v2 from 2008, v2 is **not compatible** with v1

Open source implementation available

General Network Time Transfer

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 delays are not constant → filtering required **on the client** !

Achievable accuracy does not only depend on the accuracy of the server,
It depends strongly on the **implementation** of the **client software**.
When talking about NTP or PTP distinguish between **protocol** and **implementation.**

What Affects the Network Delay?

1. The sending program puts the current time into a network packet.

2. Packet passes down the protocol stack to the network driver. Execution time depends on CPU power, interrupts, task switching → unknown delay

3. Packet has to be queued if a different packet is currently being sent → unknown delay

4. On the wire the delay is constant, depending on the cable length → constant delay

5. In routers or switches the packet can be queued for an unknown time → unknown delay

6. At the client side an interrupt is generated when the packet comes in. The interrupt handler is called to pick up the packet, unless a higher prioritized interrupt is actually in progress → unknown delay

7. Packet moves up the protocol stack to the receiving application which then takes a receive time stamp. Execution time depends on CPU power, interrupts, task switching → unknown delay

Gigabyte NICs can queue several packets before they generate an interrupt request

→increases throughput, but bad for timing applications

Network Delay Compensation

The only constant delay is the propagation delay on the wire

Take a timestamp in the NIC chip when a packet comes in from the wire or goes out onto the wire → **hardware timestamping**

Hardware timestamping can only be used if the NIC hardware, the NIC driver, the network protocol, and the time synchronization application support this

Also switches and routers need to timestamp incoming and outgoing network time transfer packets and add the measured delay to the packet contents

NTP does not support hardware timestamping but the reference implementation provides very strong filter algorithms and thus yields pretty good accuracy even over WAN connections

PTP supports hardware timestamping, and thus can yield higher accuracy than NTP, but **only** if **every node** supports hardware timestamping

Hardware Timestamping

A „Time Stamp Unit“ (TSU) listens on the wire directly at the point when a packet enters the network port and records the time when it detects a packet

The packet is passed up the network stack to the time synchronization application

The time synchronization application retrieves the hardware time stamp for this packet from the TSU

Thus execution time in the network stack has no effect on accuracy.

NTP Key Features

- High accuracy time synchronization for computers across the network, even over WAN connections
- Cross platform: originally for Unix-like systems, but ported to Windows
- Hierarchical structure with redundancy
- Optional security by authentication
- Disciplines the local system clock smoothly
- Works using IPv4 and IPv6

NTP Overview

- Initiated in the 1980's by Dave L. Mills
- Protocols and algorithms have been specified in some RFCs
- Earlier versions called XNTP, now NTP
- SNTP (Simple NTP) is a subset of NTP. Same packet format, but simpler algorithms, and thus lower accuracy than full-featured NTP.
- Works exclusively using UTC, does not care about time zones and DST which are handled by the operating systems
- Reference implementation by University of Delaware is freely available
- Current version is NTP v4
- NTP v3 is still being used
- All NTP protocol versions are compatible with earlier versions
- Actively maintained, mostly by volunteers
- Support available via NTP news group and several mailing lists
- Also Meinberg supports NTP
- NTP public services hosted by Internet Systems Consortium (ISC) which also hosts DNS name server BIND, DHCP server, Usenet news server INN
- NTP working group has been founded at IETF for additional standardization
- 3rd party implementations often lacking features

NTP Operation

- Clients receive the current time from servers
- Servers make time available to clients
- Peers which agree to a common network time which is made available to clients
- Clients only accept a server if the server claims to be synchronized
- Stratum values indicate the hierarchy level
- Local reference time sources (e.g. GPS receivers) provide the top level of time synchronization, stratum-0
- Stratum-1 servers should have direct access to a reference time source
- Every client can also act as server, a client of a stratum-n server is in turn a stratum-(n+1) server
- Every client can poll several servers for redundancy, and to detect false tickers
- NTP implementation provides efficient algorithms and filters to compensate network delay

NTP Packets on the Network

- NTP services use socket 123 which has been assigned by the Internet Assigned Numbers Autority, IANA
- Only UDP is used because TCP introduces too much overhead and jitter
- Normally clients send query packets to servers, servers send response
- For a large number of clients, server can send broadcast packets
- Broadcast packetes don't pass beyond routers
- Multicast is a special kind of broadcast where packets are sent to special multicast IP addresses; IPv4 224.0.1.1 and IPv6 ff05::101 (site local) have been assigned by the IANA
- Multicast packets can pass beyond routers
- Broadcast packets go to a single interface, multicast packets go to all interfaces
- Manycast is a way to autodetect several nearest time server using multicast techniques
- Packets can contain cryptographic signatures to check authenticity
- Poll interval from 64 to 1024 seconds, low network load

  1000 clients: ~16 requests/sec@64 sec, ~1 request/sec@1024 sec

NTP Authentication

- NTP v3 uses symmetric key cryptography:
  - Originally DES-CBC (Cipher Block Chaining)
  - Replaced by RSA MD5 (Message Digest)
  - 65534 keys can be distinguished
  - Keys and related information are specified in a key file which must be distributed and stored in a secure way
  - Besides the keys used for ordinary NTP associations, additional keys can be used as passwords for the *ntpdc* utility program
  - Individual keys must be activated with the *trusted* command before use

- NTP v4 additionally supports public key cryptography
  - Providing autokey scheme using OpenSSL
  - Autokey protocol exchanges cryptographic values in a manner designed to resist clogging and replay attacks
  - Autokey uses timestamped digital signatures to sign a session key and then a pseudo-random sequence to bind each session key to the preceding one and eventually to the signature

NTP Authentication (continued)

- Public key cryptography is more secure than symmetric key cryptography since the private key must never be revealed
- Public key management is based on X.509 certificates which can be provided by existing PKI services or produced by OpenSSL utility programs
- Authentication is configured separately for each association using the *key* or *autokey* keywords
- Authentication is always enabled, although ineffective if not configured
- NTP packets including a message authentication code are accepted only if they pass all cryptographic checks which require correct key ID, key value and message digest
- *auth* flag is enabled by default so new broadcast/manycast client, symmetric passive associations, and remote configuration commands must be cryptographically authenticated using either symmetric key or public key cryptography
- The *ntp-keygen* utility program generates several files, containing MD5 symmetric keys, RSA and DSA public keys, identity group keys, and self signed X.509 Version 3 certificates

Monitoring the NTP Status

The command `ntpq -p` can be used to monitor the time synchronization status. The example below is from a Linux server located at Meinberg in Germany, with built-in GPS PCI card, and a few configured upstream NTP servers:

```
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*SHM(0)          .GPSs.           0 l   14   16  377    0.000    0.000   0.002
+ntp-master-1.py .PPS.            1 u   10   16  377    0.165   -0.005   0.008
 ptbtime1.ptb.de .PTB.            1 u   12   64  177   11.900    0.272   0.797
 ptbtime2.ptb.de .PTB.            1 u   51   64  377   13.947   -0.170   0.319
 tick.usno.navy. .USNO.           1 u   62   64  377  108.186   -0.742   3.539
 ntp1.usno.navy. .USNO.           1 u   57   64  357  105.770    0.343   3.067
```

`ntp-master-1` is a Meinberg LANTIME on the local network. The PTB servers are located at Physikalisch-Technische Bundesanstalt (PTP) in Germany, and the USNO servers at the U.S. Naval Observatory in the United States.

The table shows that the offset can be determined with an accuracy of better than 1 millisecond even over a cross Atlantic WAN with network delay of more than 100 ms. The jitter number indicates the magnitude of the network jitter, and thus the magnitude of the potential offset estimation error.

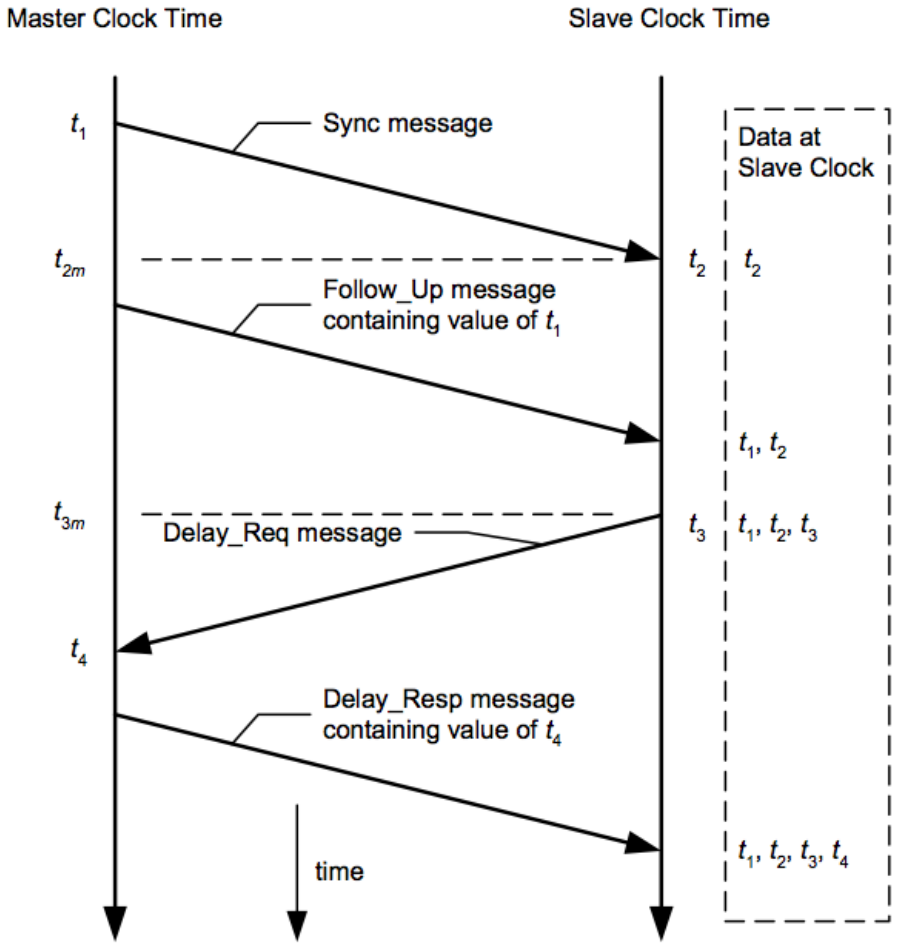
PTP: How does it work?

1. **M** → **S**: SYNC
„It's 12:03:45.6653776“

2. **M** → **S**: FOLLOW-UP
„When I said it was 12:03:45.6653776,
it was actually 12:03:45.6658994“

3. **S** → **M**: Delay Request
„Tell me when you receive this!“

4. **M** → **S**: Delay Response
„Got your last message
at 12:03:48.6726323“



PTP Characteristics

Network Layer: Layer 3 (IP), Layer 2 (Ethernet), or some Industrial bus

Modes of Operation: Ordinary, Grandmaster, Boundary Clock, Transparent Clock

Delay Mechanisms: End-to-End, Peer-to-Peer

Transmission: Multicast (default), Unicast (v2 only)

Timescale: UTC (TAI + UTC offset), Leap seconds supported

PTP Power Profile specified in IEEE C37.238

- Specifies settings to be used in Power Industry
- Adds some protocol extensions (TLVs), e.g. local time zone information

Hardware Reference Time Sources

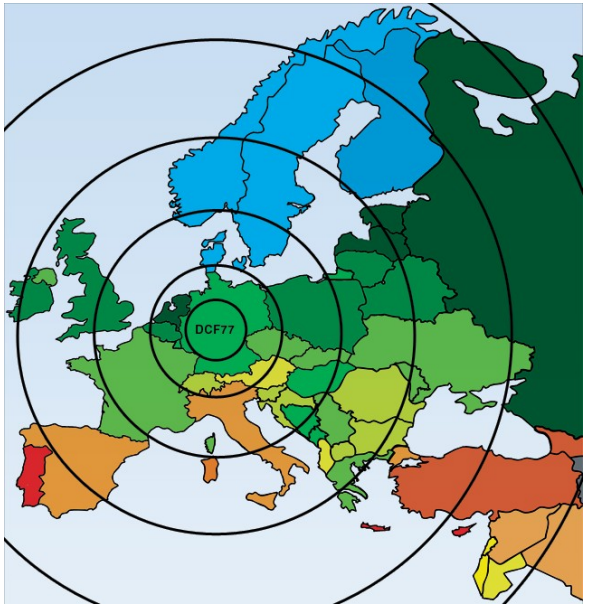
Hardware reference time sources can be used to supply same time to devices at different locations

Achievable accuracy depends on

- type of reference signal
- how accurately the time can be read from the device
- the implementation of the software
- the operating system

Longwave Transmitters

- DCF77 Germany, civil
- MSF Rugby U.K., civil
- WWVB U.S., civil
- JJY Japan, civil



- Usable only in certain regions
- Usually legal civil time
- Eventually indoor antenna possible
- Sensitive to electrical noise
- Limited accuracy

- DCF77 AM Modulation 5 ms
- DCF77 PZF Phase Modulation 50 µs

PZF accuracy only achievable with configurable propagation delay compensation
about 1 ms / 300 km

Satellite Signals

- Usable world-wide
- More accurate than longwave signals
- Outdoor antenna required
- Much more resistant against electrical noise
- Automatic propagation delay compensation

• GPS	< 1µs	USA, military
• Glonass	< 1µs	Russia, military
• Compass/Beidou	< 1µs	China, military
• Galileo	< 1µs	Europe, civil

Meinberg Time Servers

The LANTIME M-Series

Network Time Servers meeting almost <u>any</u> requirement

Multiple Form Factors:

DIN Rail, Desktop, Rackmount

High Performance from 5,000 to 10,000 req/s

Supported Protocols:

NTP v2, v3, v4
SNTP v3, v4
Time
HTTP/HTTPS
SSH
Telnet
FTP/SFTP
SNMP v1, v2c, v3
IPv4 and IPv6

Flexible and powerful notifications:

- eMail (SMTP)
- WinPopup Messages
- SNMP Traps
- Wallmount Displays (VP100)
- Alarm Relay Contact

Configuration and Management via

Web Interface, Serial Console, Telnet, SSH, SNMP

Highly customizable:

Power Supply, Holdover

Why not use Internet Servers?

- Clients do not depend on an active internet connection
- Clients do not depend on the availability of an external time server
- Internet servers may not be configured and managed well
- There are publicly accessible stratum-1 servers which distribute wrong time
- Time sync can be degraded due to capacity limits caused by hacker attacks, worms, etc.
- Internet time server have been compromised by misconfigured or badly designed clients which produced a flood of NTP queries
- Authentication is not available in most cases

Why Use Meinberg LANTIMEs?

- "Black box" time server
- Simple configuration (web browser or display)
- Event notification (Email, SNMP traps, ...)
- Manageable by SNMP
- NTP versions shipped with operating systems have often been built without refclock support
- High internal accuracy using PPS which requires special configuration/kernel support
- PTP grandmaster with hardware timestamping support

MEINBERG – The Time and Frequency Company

- 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
- 70 employees (18 R&D), one central campus in Bad Pyrmont
- approx. 70 km southwest of Hannover, Northern Germany