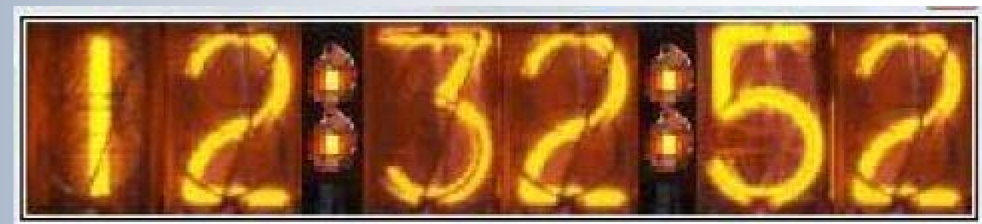
COMP 175

System Administration and Security



NETWORK TIME

Time – What's it For?

- Temporal ordering of events produced by concurrent processes
- Synchronization between senders and receivers of messages
- Coordination of joint activity
- Serialization of concurrent access for shared objects
- *"Time… is what keeps everything from happening at once".* (Ray Cummings - 1922)

Logical vs. Physical

- Logical clock keeps track of event ordering
  - Among related (causal) events
  - Relative synchrony with respect to other nodes in the distributed system

- Physical clocks keep time of day
  - Consistent across systems
  - Adhere to 'real-world' time
  - Universal Coordinated Time
    - (UTC)

Proposed project – no interest

Why

- To analyze event logs from different networked devices (routers, computers)
- To implement time-based functions, such as scheduled restarts, backups, etc.
- When multiple systems process a complex event in cooperation (database replication)
- Anything using TDM (time division multiplexing)

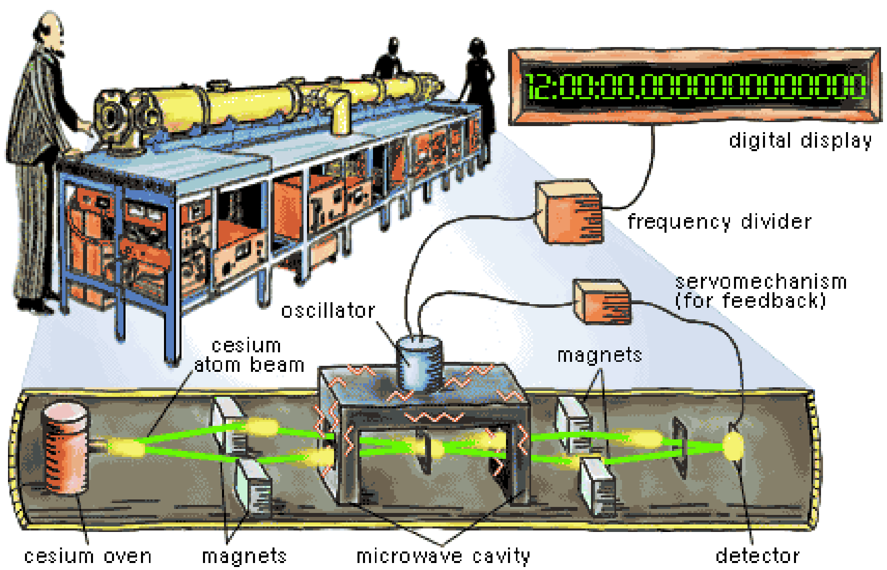
- Larger world – air, rail schedules for example
- Showing up to class/work on time

Quartz Clocks

- 1880: Piezoelectric effect
  - Curie brothers squeeze a quartz crystal & it generates an electric field
  - Apply an electric field and it bends
- 1929: Quartz crystal clock
  - Resonator shaped like tuning fork
  - Laser-trimmed to vibrate at 32,768 Hz
  - Standard resonators accurate to 6 ppm at 31°C
  - Watch will gain/lose < ½ sec/day
  - Stability > accuracy: stable to 2 sec/month
  - Good resonator - accuracy of 1 sec in 10 years
    - Frequency changes with age, temp., & acceleration

Atomic Clocks

- Second: defined as 9,192,631,770 periods of radiation corresponding to the transition between two hyperfine levels of cesium-133
- Accuracy:
  better than 1 second in six million years
- National Institute of Standards and Technology (NIST) standard since 1960



UTC

- UT0
  - Mean solar time on Greenwich meridian
  - Obtained from astronomical observation
- UT1
  - UT0 corrected for polar motion
- UT2
  - UT1 corrected for seasonal variations in Earth's rotation
- UTC
  - Civil time measured on an atomic time scale

UTC

- Coordinated Universal Time (English) CUT
- Temps Universel Coordonné (French) TUC
  - In 1970 the ITU compromised with UTC
  - Kept within 0.9 seconds of UT1
  - Atomic clocks cannot keep mean time
    - Mean time is a measure of Earth’s rotation
    - Earths revolution around its axis is slowing
    - Periodic adjustments are made

UTC and GPS

- Coordinated Universal Time (UTC)
  - International time standard
  - High precision atomic clock time
- Radio Clock
  - Receives time-coded RF signal
  - Converts to digital value
- Global Positioning System (GPS)
  - 24+ solar-powered satellites
  - Each with a primary atomic clock + backups
    - Two cesium and two rubidium atomic clocks
    - Newer phase - three rubidium atomic clocks

Clocks

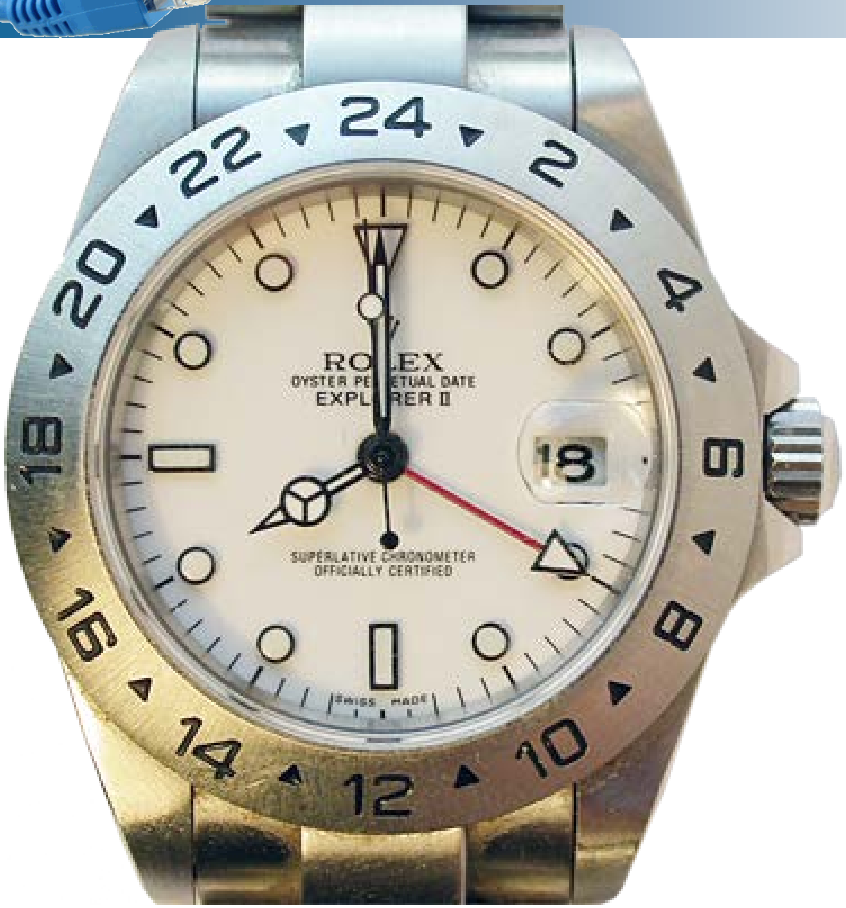
- A computer clock is an electronic device that counts oscillations at a certain frequency
  - An accurately-machined quartz crystal
- Clock skew
  - Quartz imperfections and environmental influences such as temperature, pressure, and power voltages lead to clock drift
  - Clocks run at different frequencies and will diverge from each other in a multi-node system
  - A frequency deviation of 0.001% will cause a clock to drift of ~1sec/day

Physical Clocks in Computers

- Real-time Clock: CMOS clock (counter) circuit driven by a quartz oscillator
  - Battery backup to continue measuring time when power is off
- OS generally programs a timer circuit to generate an interrupt periodically
  - e.g., 60, 100, 250, 1000 interrupts per second (Linux 2.6+ adjustable up to 1000 Hz)
  - Programmable Interval Timer Intel 8253, 8254
  - Interrupt service procedure adds 1 to a counter in memory

A Problem…

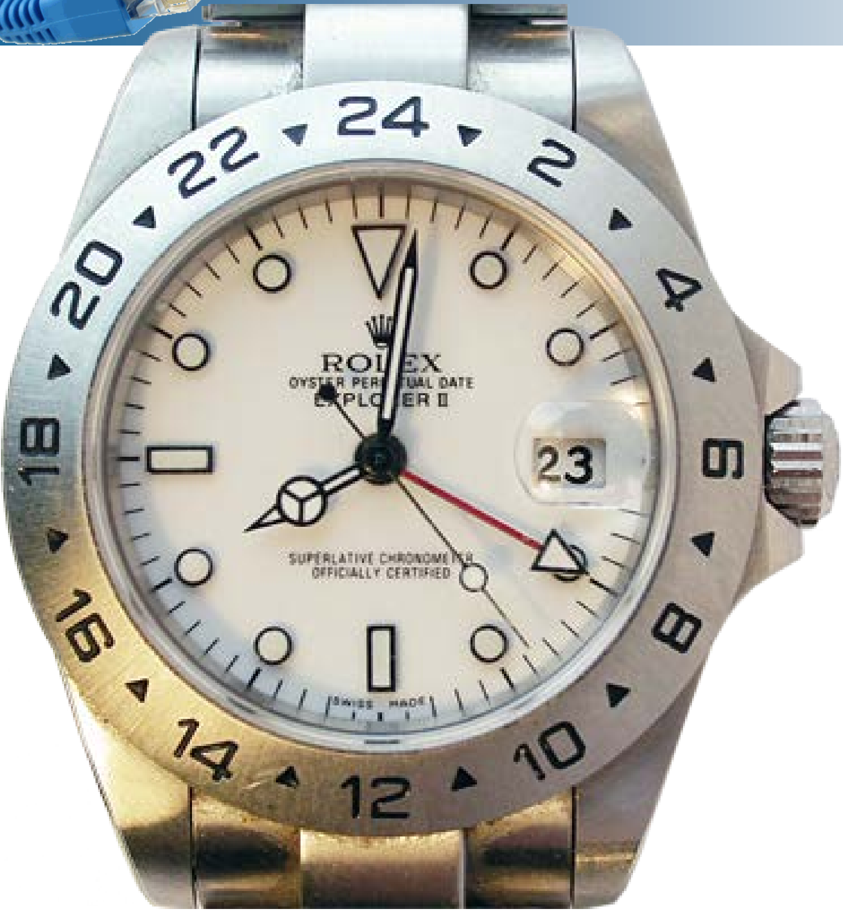
- Getting two systems to agree on time
  - Two clocks hardly ever agree
    - If one is broken – they agree once/day
  - Quartz oscillators oscillate at slightly different frequencies
- Clocks tick at different rates
  - Create ever-widening gap in perceived time
  - **Clock Drift**
- Difference between two clocks at a point in time
  - **Clock Skew**



8:00:00

8:00:00

Sept 18, 2013
8:00:00



8:01:24

Skew = +84 seconds
+84 seconds/35 days
Drift = +2.4 sec/day

Oct 23, 2013
8:00:00

8:01:48

Skew = +108 seconds
+108 seconds/35 days
Drift = +3.1 sec/day

Dealing With Drift

- Assume we set computer to true time
- Not good idea to set clock back
  - Illusion of time moving backwards can confuse message ordering and software development environments
- Go for gradual clock correction
- If fast: Make clock run slower until it synchronizes
- If slow: Make clock run faster until it synchronizes

Dealing With Drift

OS can do this:

- Change rate at which it requests interrupts
  - If system requests interrupts every 17 msec but clock is too slow, request interrupts at 15 msec
- Or software correction: redefine the interval

- Adjustment changes slope of system time:

  Linear compensating function

Resynchronizing

- After synchronization period is reached
  - Resynchronize periodically
  - Successive application of a second linear compensating function can bring us closer to true slope

- Keep track of adjustments and apply continuously
  - e.g., UNIX adjtime system call
  - ***Disciplining*** **the clock!**

Getting Accurate Time

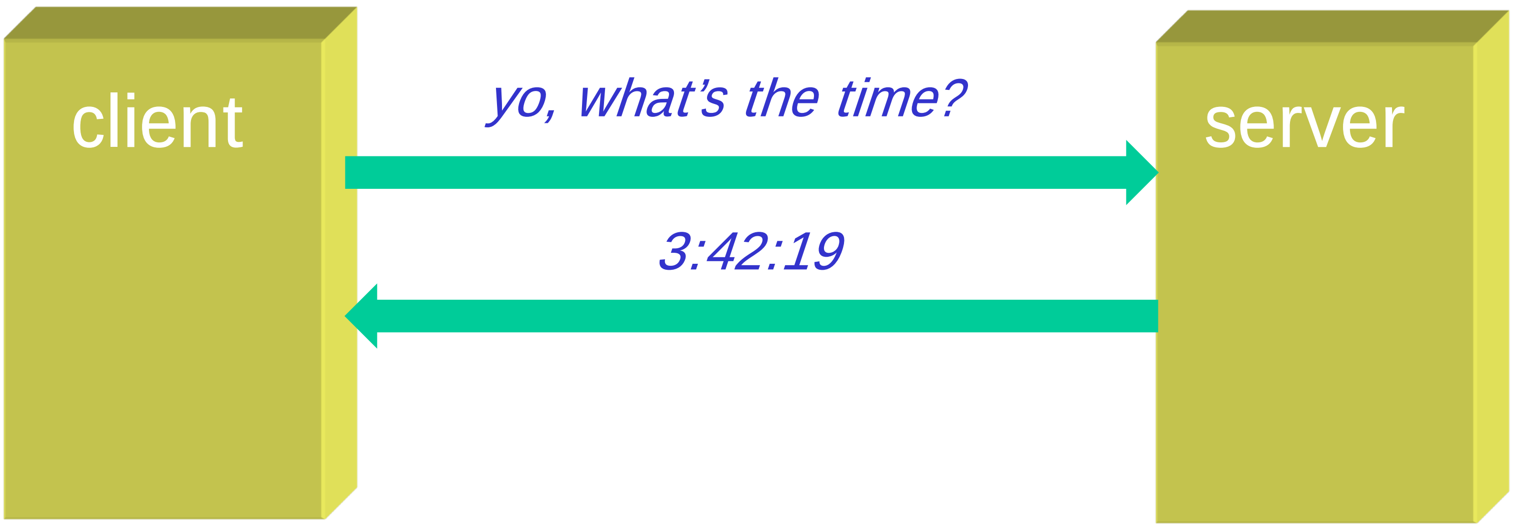
- Synchronize from another machine
  - One with a more accurate clock

- Machine/service that provides time information:
  - Time server

Remote Procedure Call

Simplest synchronization technique

- Issue RPC to obtain time
- Set time

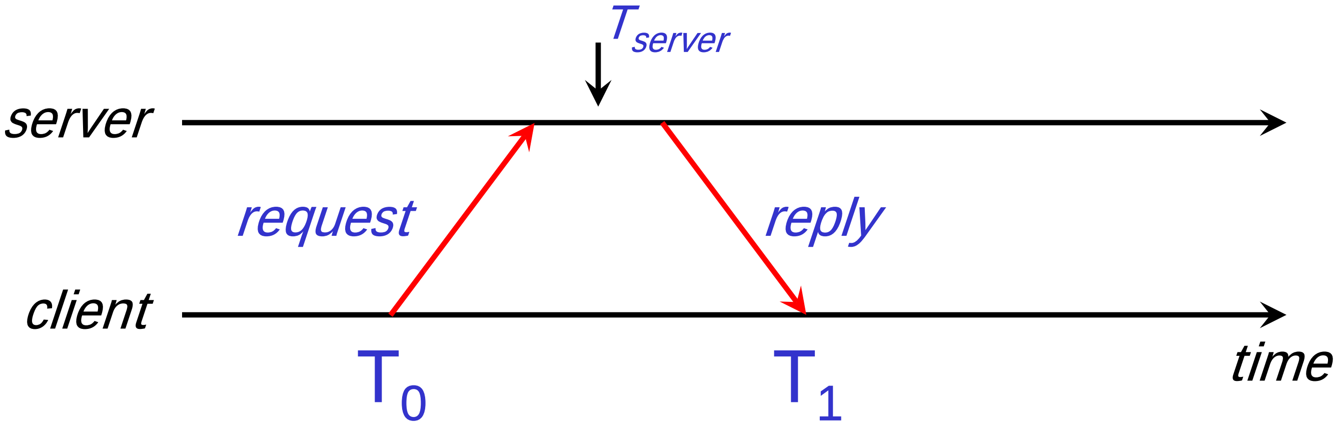


Does not account for network or processing latency

Cristian's Algorithm

Compensate for delays

- Note times:
  - request sent: T_0
  - reply received: T_1
- Assume network delays are symmetric



Cristian's algorithm: example

- Send request at 5:08:15.100 (T_0)
- Receive response at 5:08:15.900 (T_1)
  - Response contains 5:09:25.300 (T_{server})

- Elapsed time is $T_1 - T_0$

  5:08:15.900 - 5:08:15.100 = 800 msec
- Best guess: timestamp was generated

  400 msec ago
- Set time to T_{server} + *elapsed time*

  5:09:25.300 + 400 = 5:09.25.700

What time is it really?

"A man with a watch knows what time it is. A man with two watches is never sure."

- Assumes no machine has an accurate time source
- Obtain average from participating computers
- Synchronizes all clocks to average

Berkeley Algorithm

- Machines run time dæmon
  - Process that implements protocol
- One machine is elected (or designated) as the server (master)
  - Others are slaves

Berkeley Algorithm

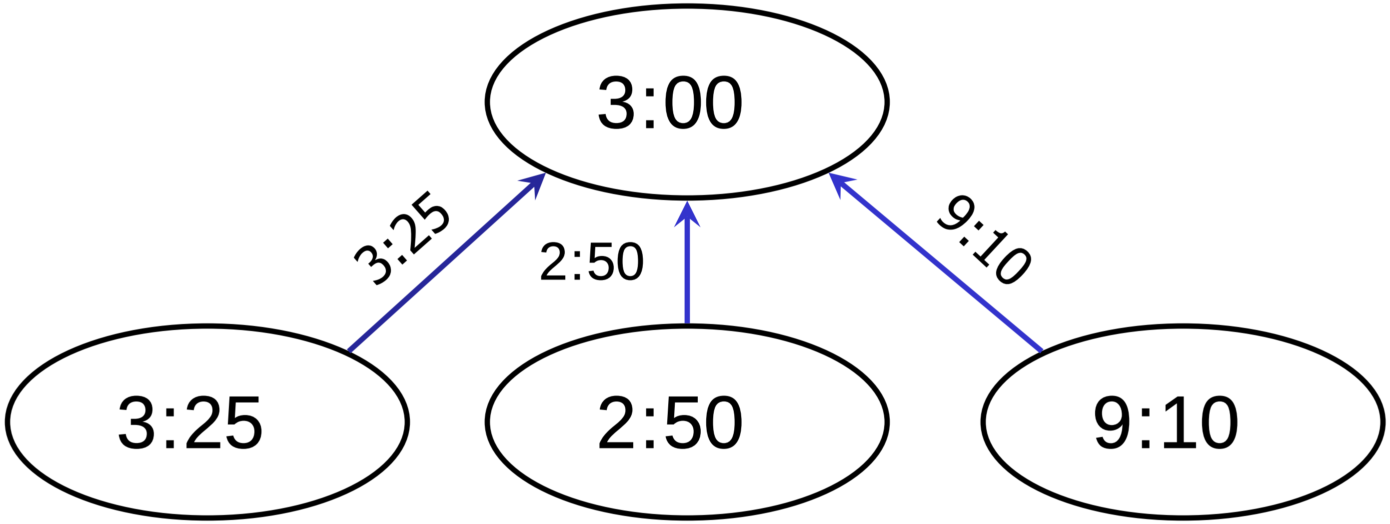
- Master polls each machine periodically
  - Ask each machine for time
    - Can use Cristian's algorithm to compensate for network latency
- When results are in, compute average
  - Including master's time
- Hope: average cancels out individual clock's tendencies to run fast or slow
- Send offset by which each clock needs adjustment to each slave
  - Avoids problems with network delays if we send a time stamp

Berkeley Algorithm

- Algorithm has provisions for ignoring readings from clocks whose skew is too great
  - Compute a fault-tolerant average

- If master fails
  - Any slave can take over

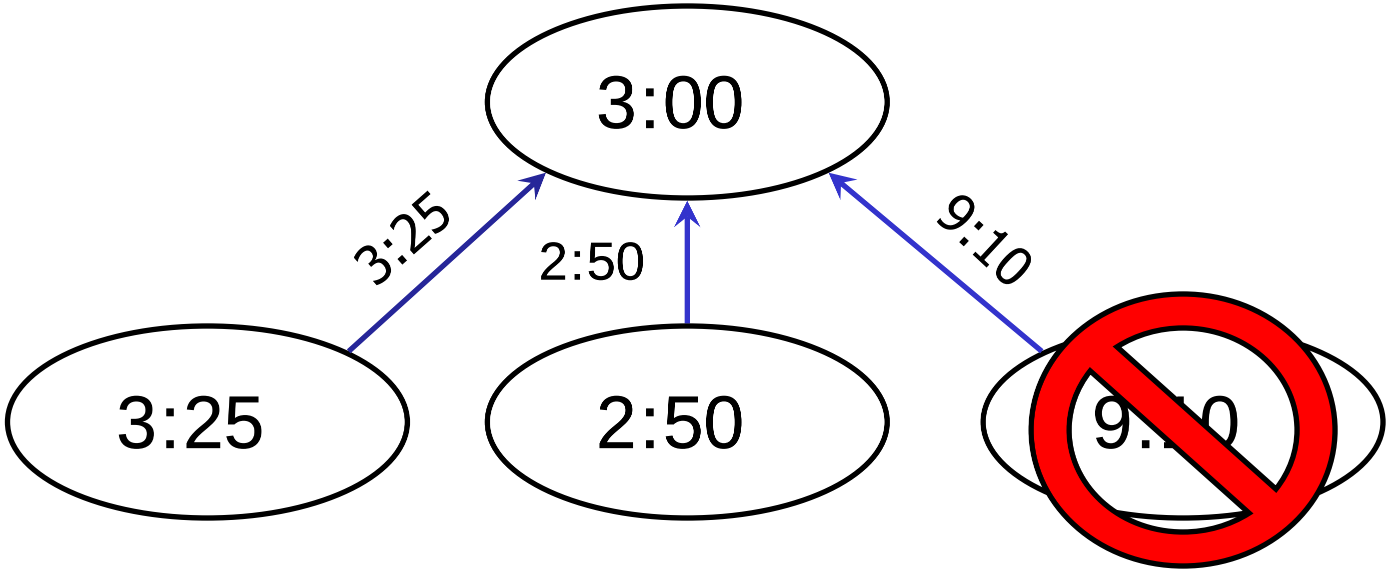
Berkeley Algorithm: example



2. Compute fault-tolerant average:

$$\frac{3{:}25 + 2{:}50 + 3{:}00}{3} = 3{:}05$$

Berkeley Algorithm: example



2. Compute fault-tolerant average:

$$\frac{3{:}25 + 2{:}50 + 3{:}00}{3} = 3{:}05$$

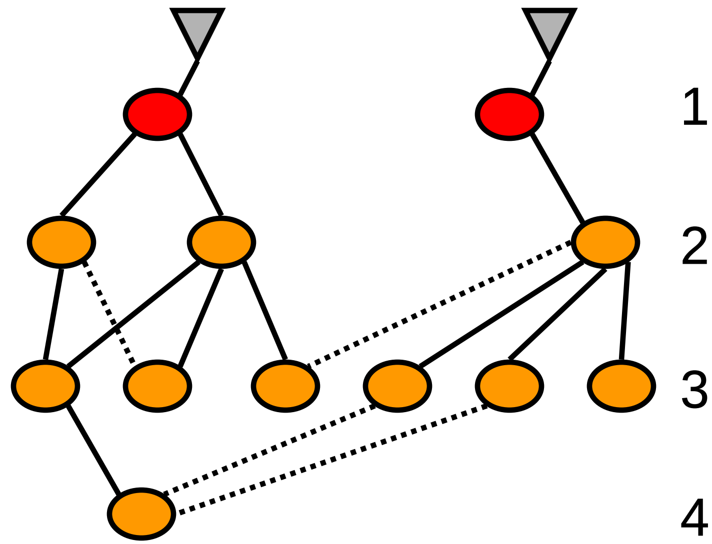
NTP Goals

- Enable clients across Internet to be accurately synchronized to UTC despite message delays
  - Use statistical techniques to filter data and gauge quality of results
- Provide reliable service
  - Survive lengthy losses of connectivity
  - Redundant paths
  - Redundant servers
- Enable clients to synchronize frequently
  - offset effects of clock drift
- Provide protection against interference
  - Authenticate source of data

NTP servers

Arranged in strata

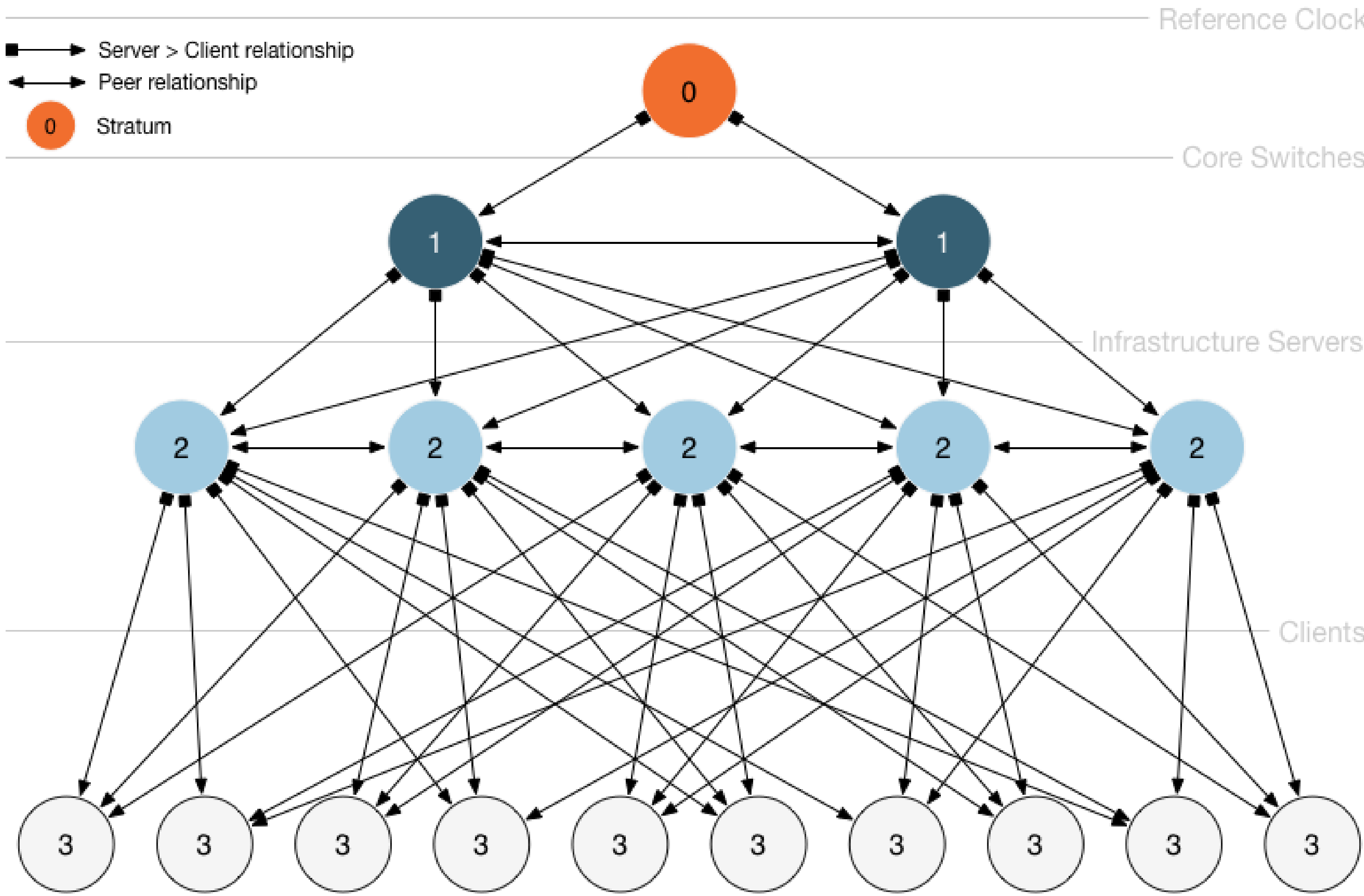
- 1^{st} stratum: machines connected directly to accurate time source
- 2^{nd} stratum: machines synchronized from 1^{st} stratum machines
- …



SYNCHRONIZATION SUBNET

Each NTP node has a stratum
Stratum is an integer between 0 and 16, inclusively
Stratum 0 means a physical clock, never a computer
Stratum 16 – pretty much worthless time-wise

NTP Architecture



NTP Synchronization Modes

- Multicast mode
  - for high speed LANS
  - Lower accuracy but efficient
- Procedure call mode
  - Similar to Cristian's algorithm
- Symmetric mode
  - Intended for master servers
  - Pair of servers exchange messages and retain data to improve synchronization over time

- All messages delivered unreliably with UDP

NTP Messages

- Procedure call and symmetric mode
  - Messages exchanged in pairs
- NTP calculates:
  - Offset for each pair of messages
    - Estimate of offset between two clocks
  - Delay
    - Transmit time between two messages
  - Filter Dispersion
    - Estimate of error – quality of results
    - Based on accuracy of server’s clock and consistency of network transit time
- Use this data to find preferred server:
  - lower stratum & lowest total dispersion

NTP Message Structure

- Leap second indicator
- Version number
- Mode (symmetric, unicast, broadcast)
- Stratum (1=primary reference, 2-15)
- Poll interval
  - Maximum interval between 2 successive messages, nearest power of 2
- Precision of local clock
  - Nearest power of 2

NTP Message Structure

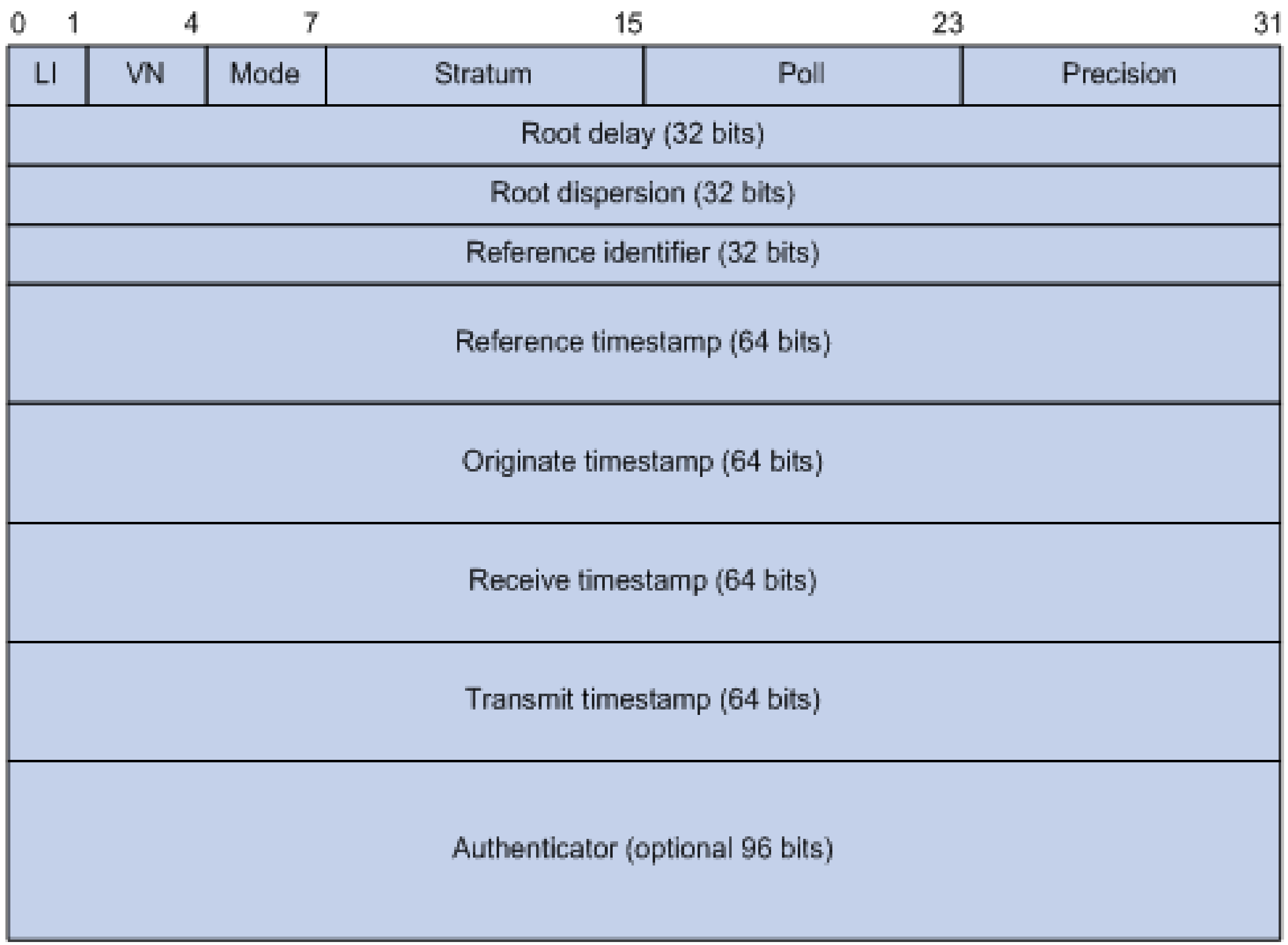
- Root delay
  - Total roundtrip delay to primary source
  - (16 bits seconds, 16 bits decimal)
- Root dispersion
  - Nominal error relative to primary source
- Reference clock ID
  - Atomic, NIST dial-up, radio, LORAN-C navigation system, GOES, GPS, ...
- Reference timestamp
  - Time at which clock was last set (64 bit)
- Authenticator (key ID, digest)
  - Signature (ignored in SNTP)

NTP Message Structure

- T1: originate timestamp
  - Time request departed client (client's time)
- T2: receive timestamp
  - Time request arrived at server (server's time)
- T3: transmit timestamp
  - Time request left server (server's time)

Message Format

LI: 2-bit leap indicator
VN: 3-bit version
Mode: 3-bit mode

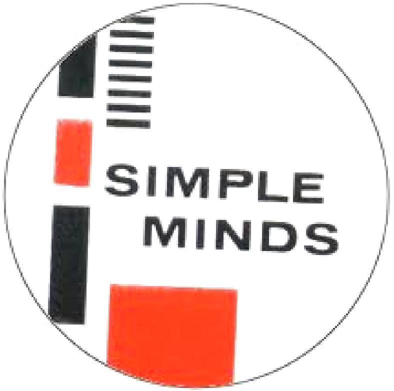


NTP's Validation Tests

- Timestamp provided ≠ last timestamp received
  - Duplicate message?
- Originating timestamp in message consistent with sent data
  - Messages arriving in order?
- Timestamp within range?
- Originating and received timestamps ≠ 0?
- Authentication disabled? Else authenticate
- Peer clock is synchronized?
- Don't sync with clock of higher stratum #
- Reasonable data for delay & dispersion

SNTP

- Simple Network Time Protocol
  - Based on Unicast mode of NTP
  - Subset of NTP, not new protocol
  - Operates in multicast or procedure call mode
  - Recommended for environments where server is root node and client is leaf of synchronization subnet
  - Root delay, root dispersion, reference timestamp ignored
  - Default Windows daemon



NTP Deployments

- An estimated 25 million Network Time Protocol (NTP) servers and clients are deployed all over the world.
- NTP software has been ported to almost every workstation and server platform available today - from PCs to supercomputers and embedded systems, even home routers and battery backup systems.
- NTP is arguably the longest running, continuously operating, ubiquitously available protocol in the Internet (since 1979)

NTP Uses

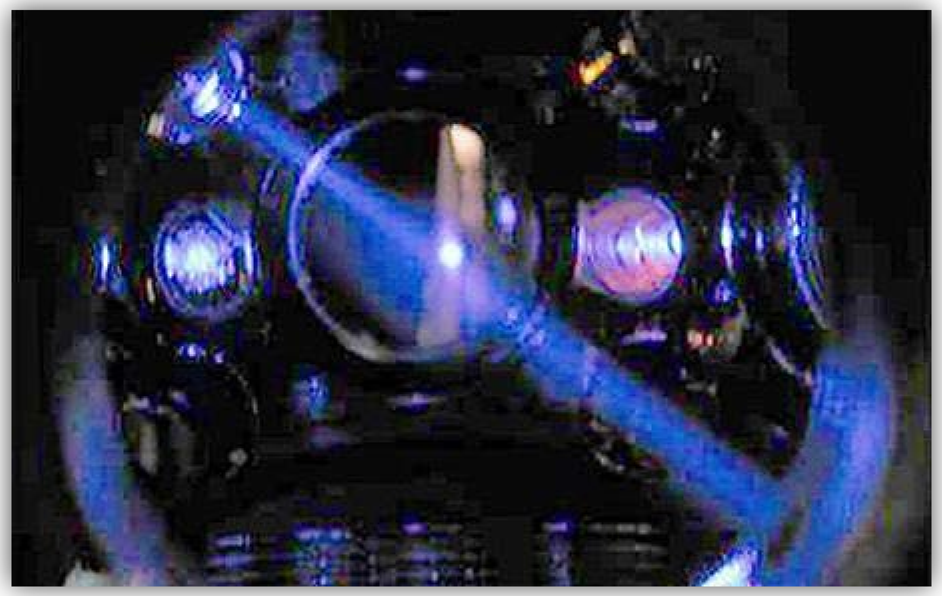
- Synchronizing global clocks with millisecond accuracy enables
  - The exact incidence of global events to be accurately determined.
  - Detection and prosecution of distributed denial of service attacks
  - Real time synchronization of applications such as multimedia conferencing.

Limitations

- NTP does not deal with time zones (that's a time display problem)
- Needs external servers to work well, even with a local clock
- Can produce systematic errors with asymmetric paths
- Can have problems with asymmetric congestion
- Do not mix strata

Accuracy

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



CMDA?

- Code-Division Multiple Access
- Data and voice packets are separated using codes and then transmitted using a wide frequency range. Bias towards data.
- 3G
- Being phased out (Verizon ends 12/31/22)
- But....GPS requires a 'sky view' antenna
  - Difficult to obtain in data centers
  - Easy in Baun Hall display case

Time Serving Hazards

- With potential client populations in the millions, a very real vulnerability to grossly overload the public primary server population and/or the interconnecting networks.
- The public NTP client software exchanges packets with the server on a continuous low-rate basis in order to discipline the computer clock time and frequency.
- The sheer weight of numbers threatens to overwhelm at least some of the current NIST and USNO servers.
- Other incidents reveal really bad network engineering and counterproductive
- Defective NTP client implementations have appeared that exhibit gross violations of the Internet social contract.
- An example is the U Wisconsin incident reported in the next slide. parameter selection, especially poll interval.

University of Wisconsin

- Operates multiple time servers for campus access
- A home router came on the market that
  - had the address of one of these servers hard-coded in firmware and could not be changed,
  - could send packets continuously at one-second intervals when the path or server was unavailable..
- Not a problem if router sales numbers are low
  - 750,000 routers were sold, most could not be recalled, updated or even reliably found
  - Resulting traffic overwhelmed the server, university network and service provider
- No elegant solution other than to insure continuous service and educate manufacturer about socially responsible product design

Conditions at USNO

- USNO operates ~20 NTP servers in U.S.
- 3 busiest servers are in Washington, DC.
- Shared aggregate load 3,000-7,000 PPS
- 3 servers running at 10% of capacity, but network is badly overloaded, leading to significant packet loss and badly degraded time quality
- Much of traffic - clients sending at unrealistic rates
- One client spraying at 14 PPS, a rate equivalent to 731 properly configured NTP clients
- One university firewall channeled 2,000 campus clients separately to USNO servers. It should synchronize to USNO and have all campus clients synchronize to it, as NTP is designed to do.

Conditions at NIST

- NIST operates ~12 NTP public time servers in US
  - Three of the busiest servers are in Boulder, CO
  - Shared aggregate load similar to USNO, but NIST network infrastructure more resilient
- An experiment collected statistics in a 9 second window on each machine captured 13% of polls
- Results: 500+ clients with polling intervals of ≤5 seconds, 15 with poll intervals < 1 sec. vs. stable NTP client rates of 14 minute intervals.
- Most incidences involved packet bursts
- One particularly offensive elephant was sending continuously at two packets per second

Configuration

```
# /etc/ntp.conf:  Configuration file for ntpd.
# run ntpdate -b <timeserver> to sync system's time initially
# internal fallback of last resort is the local clock
server 127.127.1.0 # local clock
fudge 127.127.1.0 stratum 10
#
server 216.218.254.202
server svtick.time.com
server svtock.time.com
authenticate no
#
# Drift file.  Point to directory which the daemon can write to
driftfile /etc/ntp/drift
#
logfile /var/log/ntp.log
```

This config points to 3 stratum one servers and offers stratum two to its network. You could also point to the ntp.pool servers for time.

Monitoring

```
# ntpq -p
 remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 LOCAL(0)        LOCAL(0)    10 l   43   64   377    0.000    0.000   0.008
*clock.sjc.net   .CDMA.       1 u  262 1024   377   49.338   -3.298   0.504
 svtick.time.com .CDMA.      16 u 276m  512     0    0.000    0.000 4000.00
+svtock.time.com  .GPS.       1 u  197 1024   377   57.816   -2.455   1.434
```

Remote the ntp server you specified in your configuration

Refid the ntp server time source

St (Stratum) how far is the server from the time source – in stratum

T (type) type of source, u:unicast b:broadcast/multicast s:symmetric peer

When the number of seconds passed since last response

Poll the polling interval in seconds.

Reach a bit vector (in octal) of the last eight poll attempts (1=replied, 0=no reply)

Delay the roundtrip time to receive a reply, in milliseconds.

Offset time difference between client server and source, in milliseconds

Jitter the difference between two samples, still in milliseconds

Monitoring

```
# ntpq -p
 remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 LOCAL(0)        LOCAL(0)    10 l   43   64  377    0.000    0.000   0.008
*clock.sjc.net   .CDMA.       1 u  262 1024  377   49.338   -3.298   0.504
 svtick.time.com .CDMA.      16 u 276m  512    0    0.000    0.000 4000.00
+svtock.time.com  .GPS.       1 u  197 1024  377   57.816   -2.455   1.434
```

Unreachable for 276 minutes, svtick was moved to stratum 16, essentially discarded. ntpq is one of the utilities included in a full ntp implementation.

Sources – two are CDMA (cell infrastructure)
one is GPS

Discussion

- How many time sources are needed?

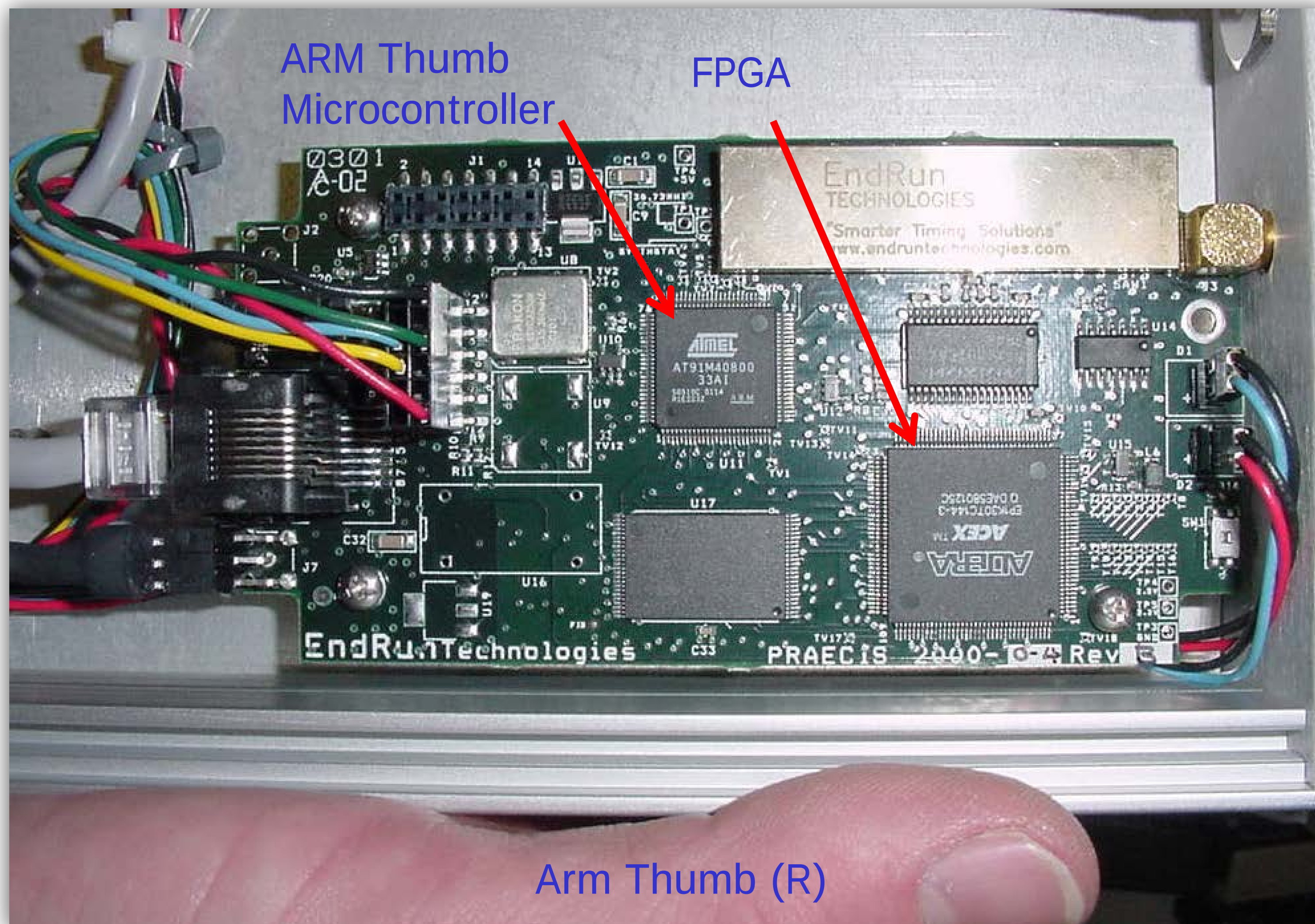
- 1 ?
- 2 ?
- 3 ?
- 4 ?
- 5 ?

Discussion

- How many time sources are needed?

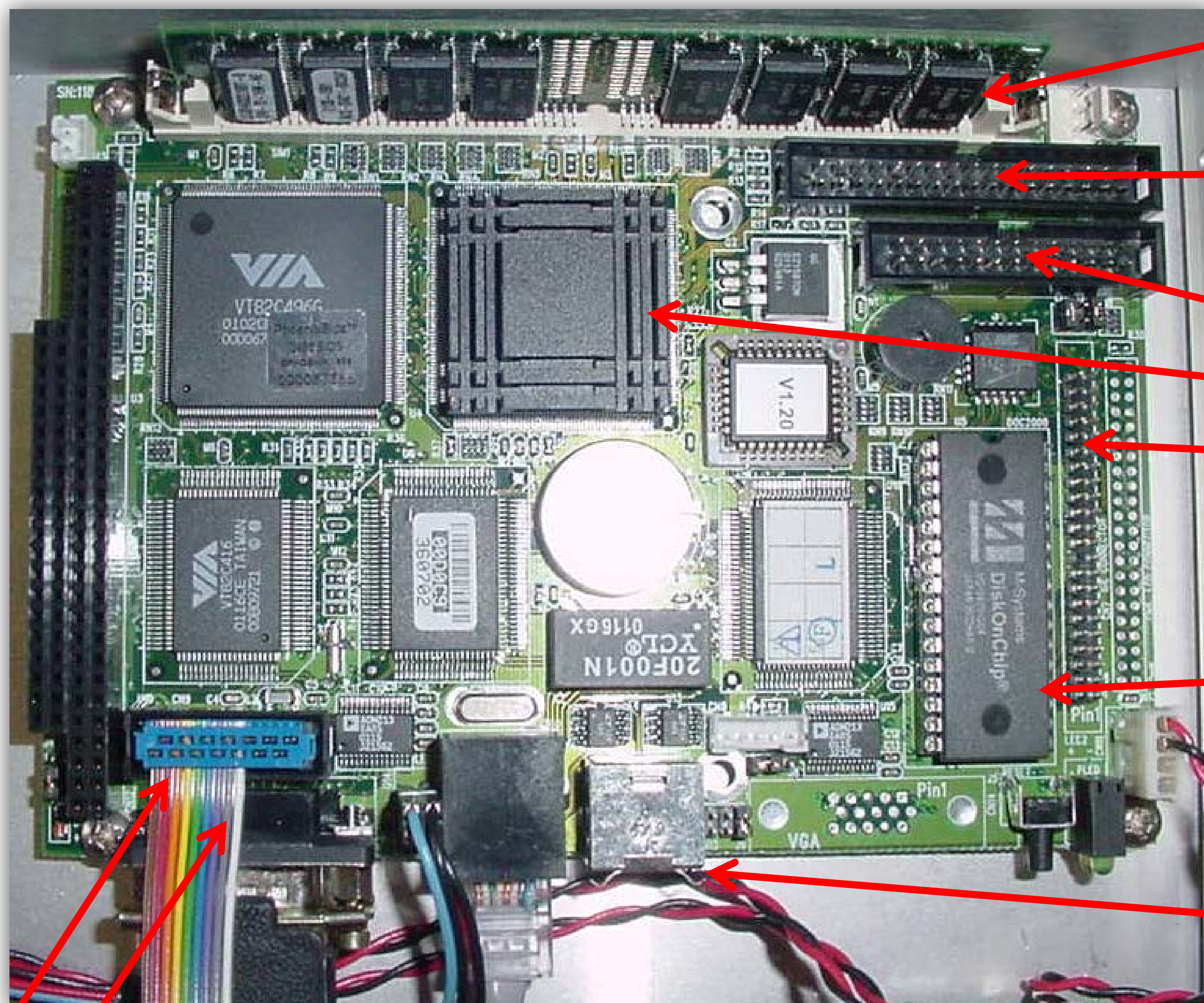
- 1 ?
- 2 ?
- 3 ?
- 4 ?
- 5 ?

Stratum 0 Time Reference



CDMA Time Server – The receiver card 850MHz

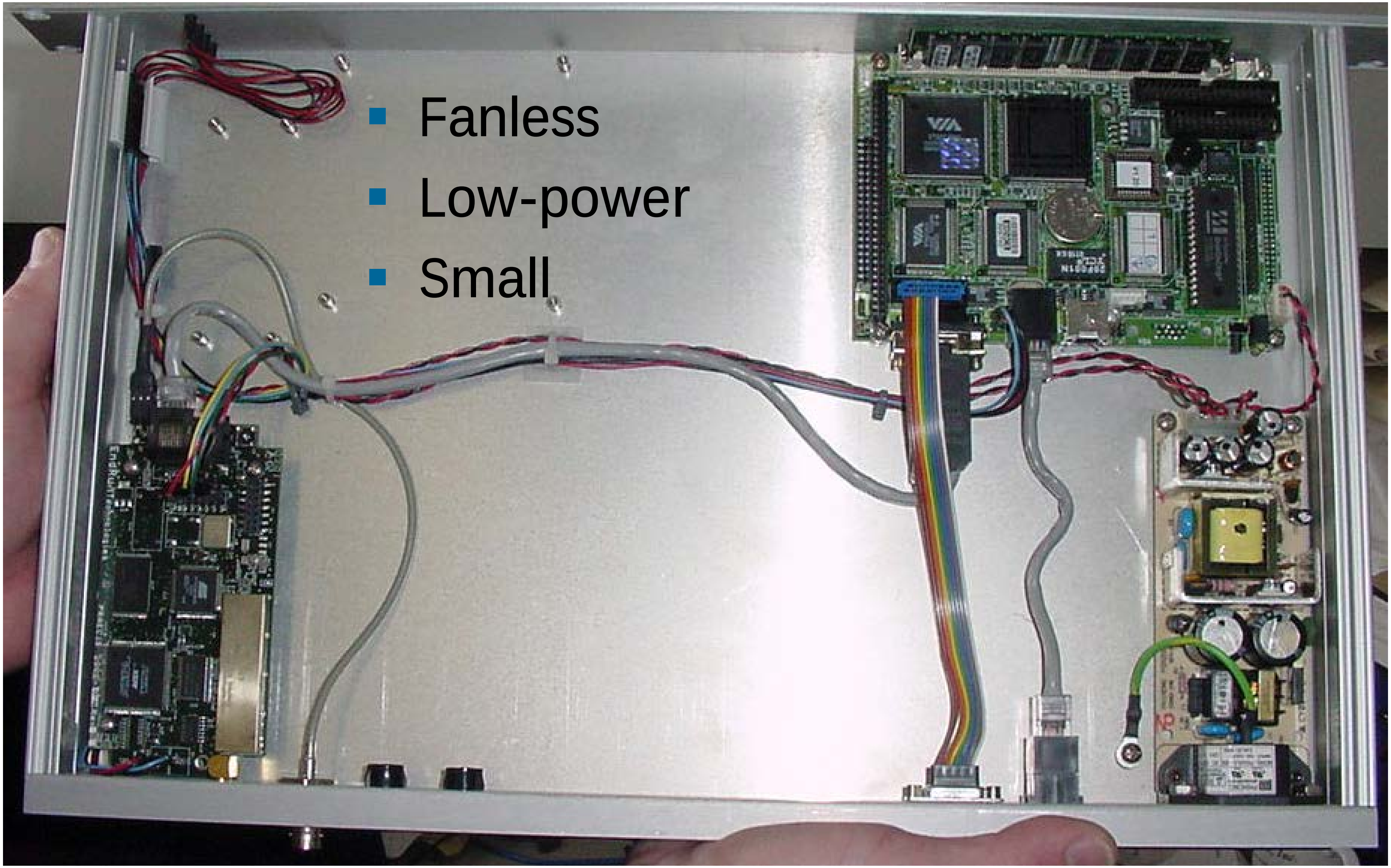
Stratum 1 Server



Advantech PCM-4825 5x86 133MHz

1U Rack Enclosure

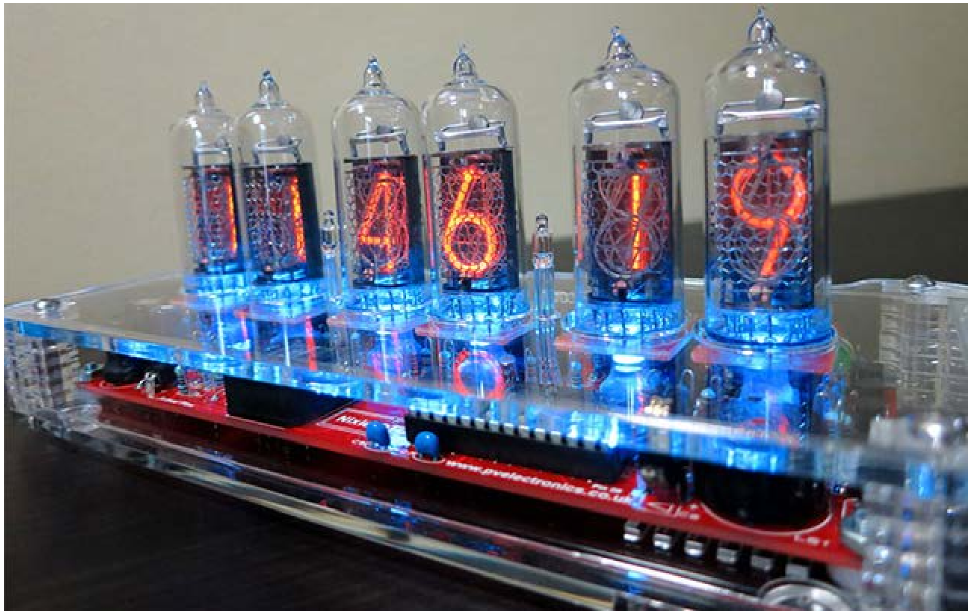
- Fanless
- Low-power
- Small



Sources

- WWW – radio
- CDMA - Code division multiple access (CDMA)
- GPS – satellite

- Go look in the Baun lobby



Leap Seconds

- A leap second is a one-second adjustment
  - Last minute has 59, 60, 61 seconds
- Occasionally applied to UTC
- To keep UTC close to the mean solar time
- Irregularly spaced and unpredictable
- Most recent leap second insertion
  - December 31, 2016 at 23:59:60 UTC

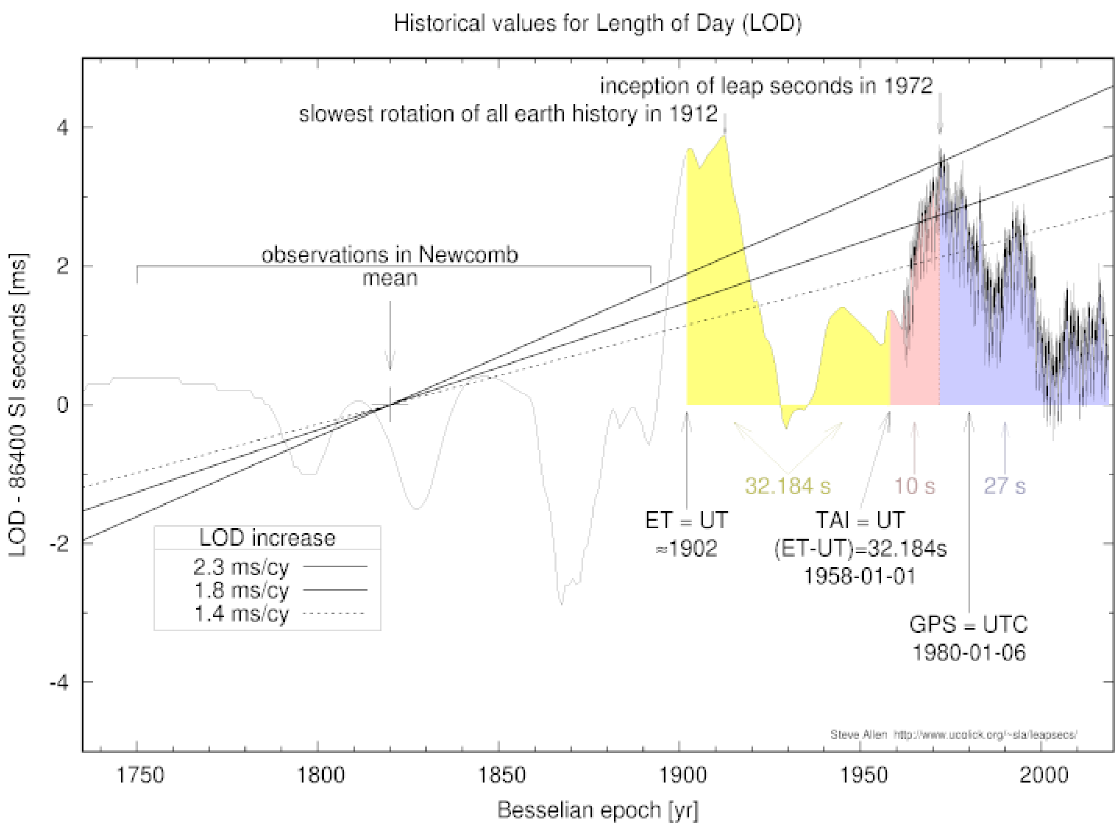
dmesg

dmesg:

Clock: inserting leap second 23:59:60 UTC

Length Of Day (LOD)

- Over the past 30 years the rotation of the earth's crust accelerated due to changes of fluid circulation in the outer core of the earth.



Length Of Day

- Earths rotation slowing due to tidal influences
- Using LOD ~ 1820 and calculate tidal deceleration
  - 2 leap sec/yr by 2142
  - 4 leap sec/yr by 2464
  - 12 leap sec/yr by 3753

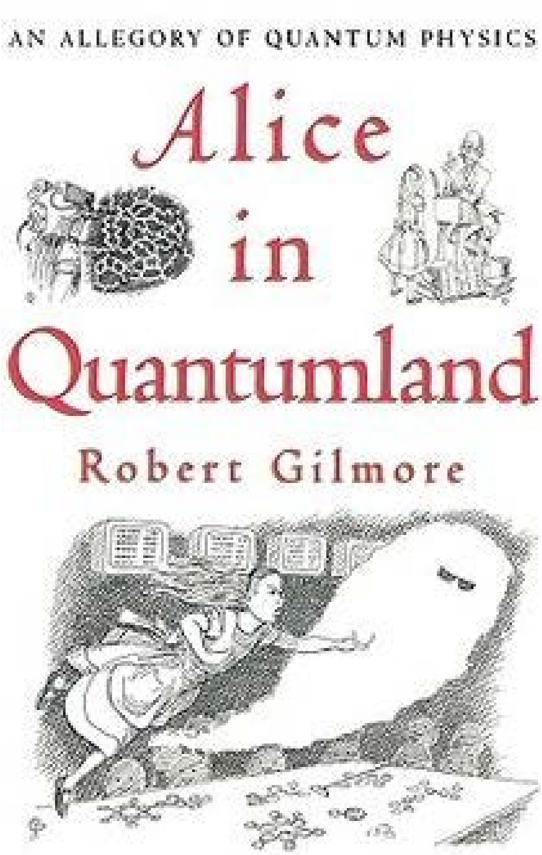
- Current system should work for next 1200 years.
- Unless civilization decides that counting days and nights is irrelevant, civil time will want to remain synchronized with solar time, not atomic time.

Earth Spins

- Earth spun on its axis a bit faster June 29, 2022
- Shortest days since measurements began in 60's
- Earth day was 1.59 milliseconds shy of 24 hours
- There were 28 short days in 2020
- Causes:
  - Earthquakes
  - El Niňo
  - Moons gravity
  - Storms
  - Vanishing glaciers

Time

- A physicist, an astronomer, and a satellite technician are standing at a street corner. A passerby comes along and asks: "Pardon me, what time is it?"



Resources

www.ntp.org/ The NTP Project

www.eecis.udel.edu/~mills/exec.html NTP Executive Summary

www.eecis.udel.edu/~mills/index.html NTP Creator's Page

www.endruntechnologies.com Vendor

www.symmetricom.com Vendor

Google for the current location of the Sun/Oracle Blueprint series on NTP

norloff.org/ntp/ replacements for the sntp Windows client

Remember

- NTP keeps Coordinated Universal Time (UTC) ~~GMT~~
  - UTC source is atomic clock
- Leap seconds correct for Earth rotation issues (e.g. mean solar time – aka UT0)
- Clock Drift - ever-widening gap in perceived time
- Clock skew – variance between two clocks
- Applying continuous adjustments to clock is disciplining the clock
- Offset - The difference between the time reported by a clock and the true time as defined by UTC
- A truechimer is a clock that maintains timekeeping accuracy to a previously published (and trusted) standard
- A falseticker is a clock that does not