Network Time Protocol (NTP) Recent Developments

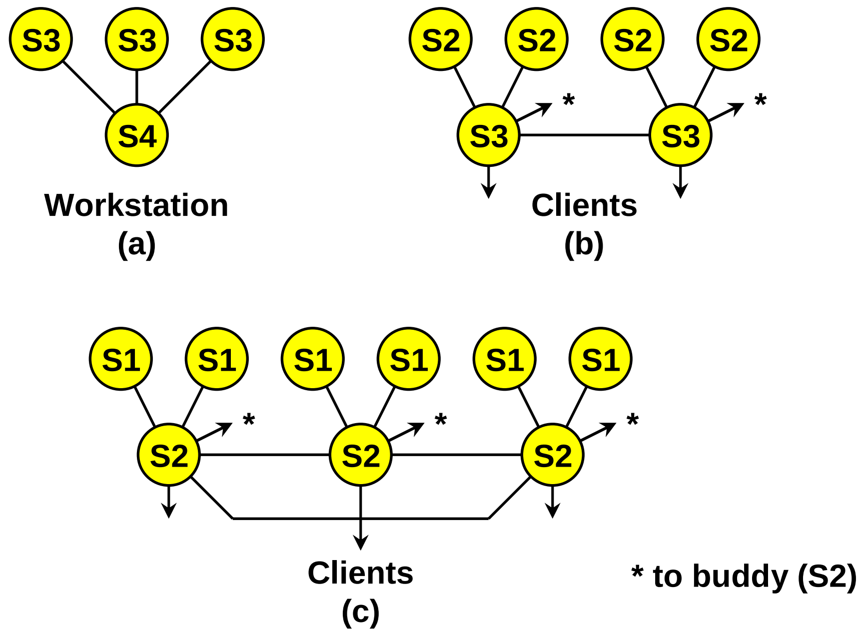
Sir John Tenniel; *Alice's Adventures in Wonderland,*Lewis Carroll

David L. Mills
University of Delaware
http://www.eecis.udel.edu/~mills
mills@udel.edu

Introduction

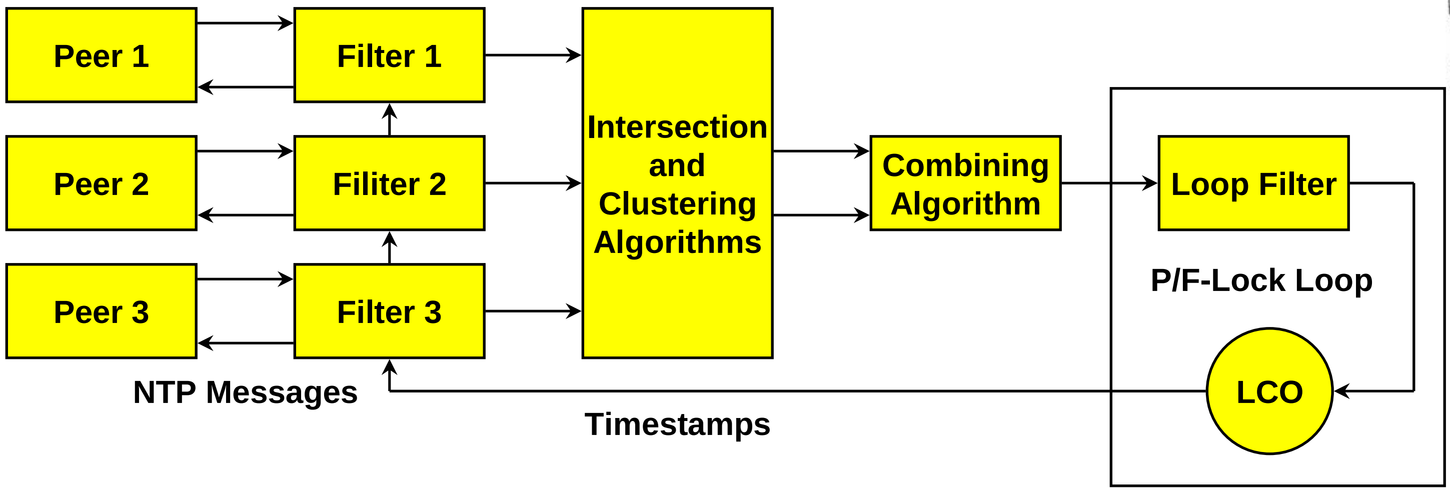
- Network Time Protocol (NTP) synchronizes clocks of hosts and routers in the Internet
- Provides submillisecond accuracy on LANs, low tens of milliseconds on WANs
- Primary (stratum 1) servers synchronize to UTC via radio, satellite and modem; secondary (stratum 2, ...) servers and clients synchronize via hierarchical subnet
- Reliability assured by redundant servers and diverse network paths
- Engineered algorithms used to reduce jitter, mitigate multiple sources and avoid improperly operating servers
- Unix NTP daemon ported to almost every workstation and server platform available today - from PCs to Crays
- Well over 100,000 NTP peers deployed in the Internet and its tributaries all over the world

NTP configurations



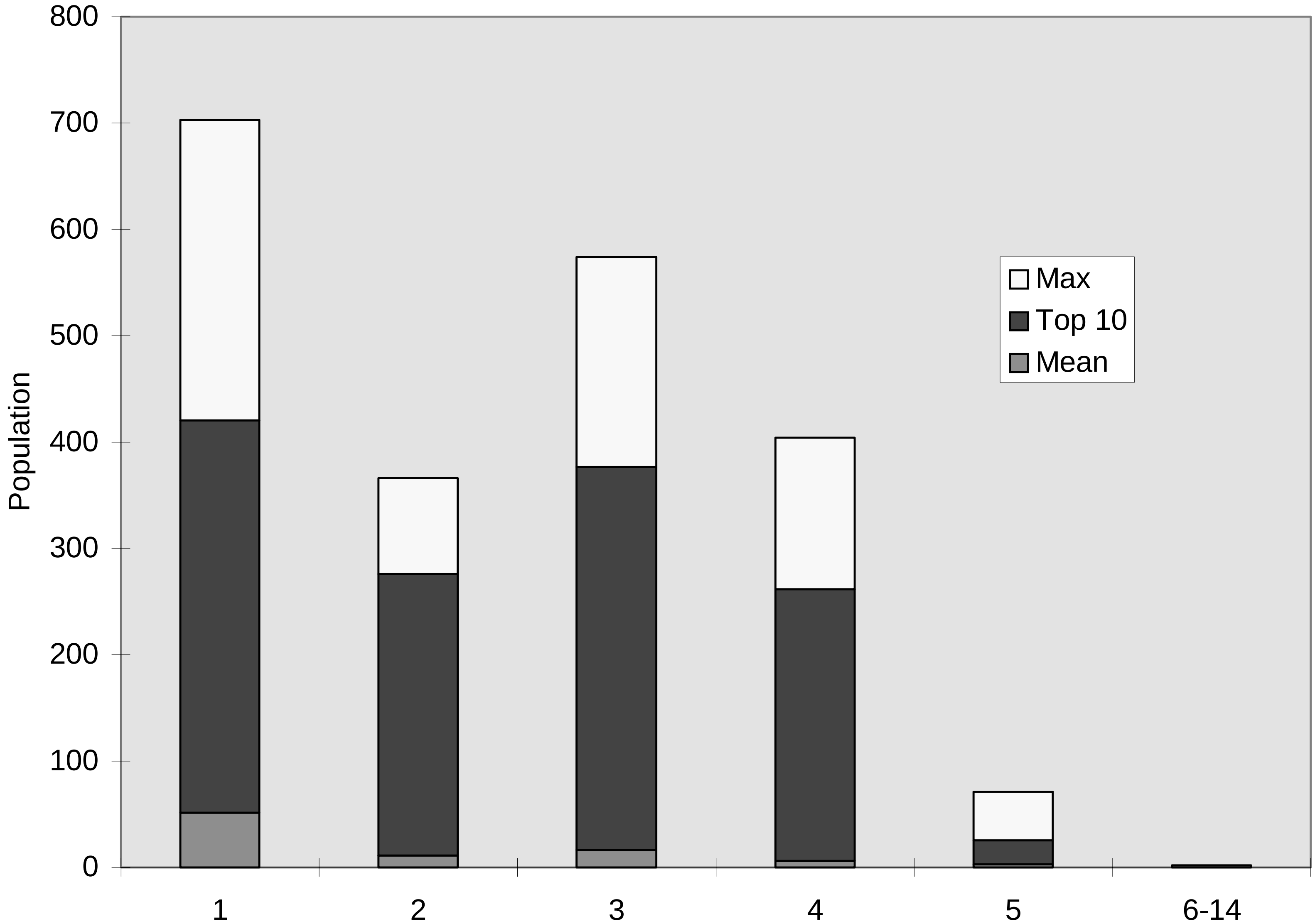
- (a) Workstations use multicast mode with multiple department servers
- (b) Department servers use client/server modes with multiple campus servers and symmetric modes with each other
- (c) Campus servers use client/server modes with up to six different external primary servers and symmetric modes with each other and external secondary (buddy) servers

NTP architecture

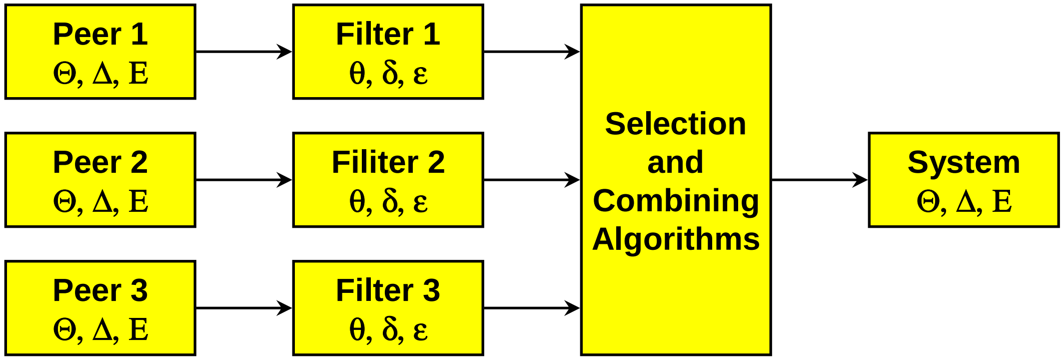


- Multiple synchronization peers for redundancy and diversity
- Data filters select best from a window of eight clock offset samples
- Intersection and clustering algorithms pick best subset of peers and discard outlyers
- Combining algorithm computes weighted average of offsets for best accuracy
- Loop filter and local clock oscillator (LCO) implement hybrid phase/frequency-lock feedback loop to minimize jitter and wander

Clients per server population by stratum

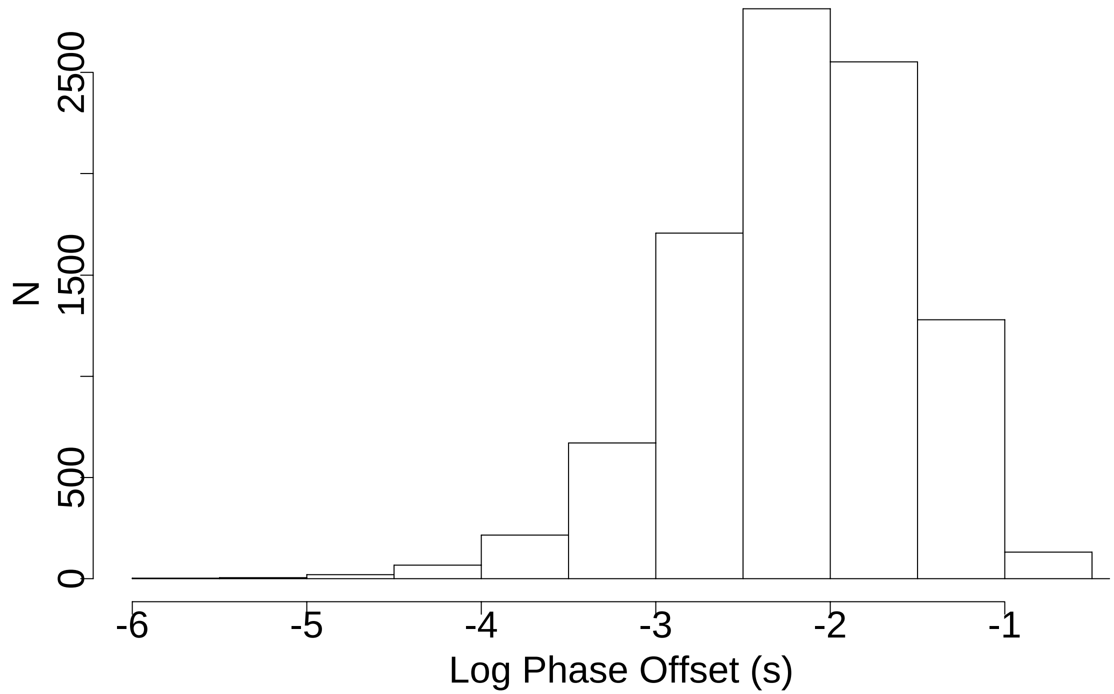


Comprehensive error budget



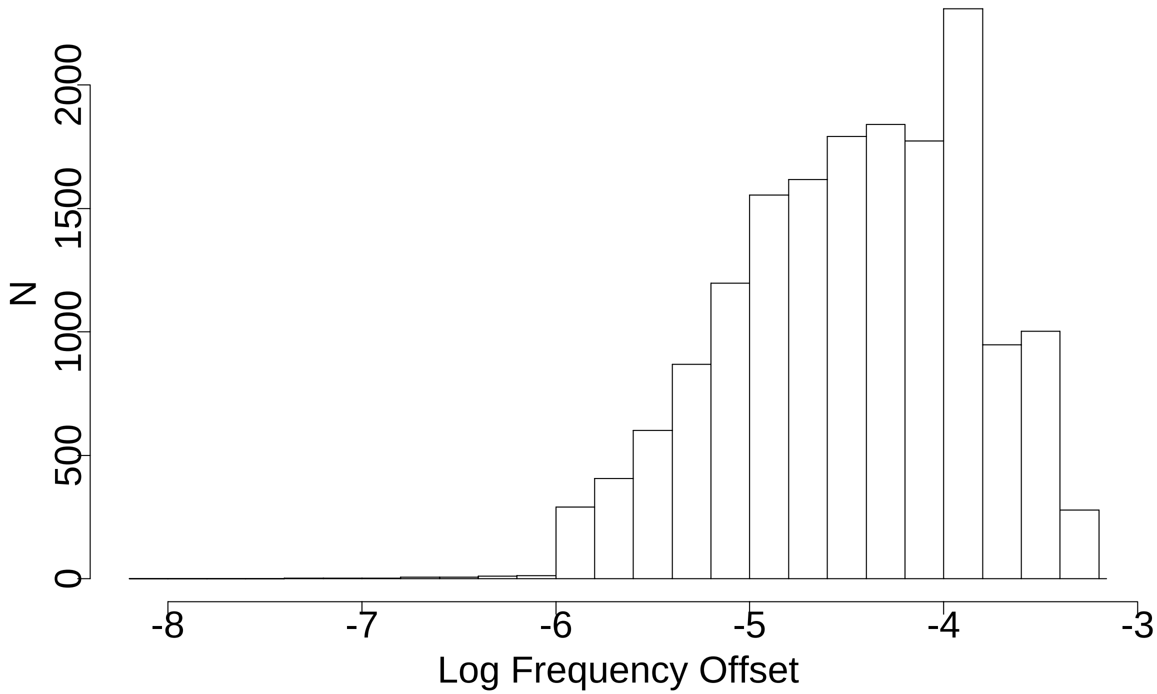
- Error budget is recalculated at each NTP message arrival
- Each peer calculates offset Θ, delay Δ and dispersion E relative to the root of the synchronization subtree
- Client estimates increments θ, δ, ε in clock filter algorithm
- Clock selection and combining algorithms combine and weight the sums to construct new Θ, Δ, E for the client
- Dispersions increase with time at rate equal to specified frequency tolerance

Local clock phase offsets compared



- Histogram of local clock absolute phase offsets
  - 19,873 Internet peers surveyed running NTP Version 2 and 3
  - 530 offsets equal to zero deleted as probably unsynchronized
  - 664 offsets greater than 128 ms deleted as probably unsynchronized
  - Remaining 18,679 offsets: median 7.45 ms, mean 15.87 ms

Local clock frequency offsets compared



- Histogram of local clock absolute frequency offsets
  - 19,873 Internet peers surveyed running NTP Version 2 and 3
  - 396 offsets equal to zero deleted as probably spurious (self synchronized)
  - 593 offsets greater than 500 PPM deleted as probably unsynchronized
  - Remaining 18,884 offsets: median 38.6 PPM, mean 78.1 PPM

A day in the life of a busy NTP server

- NTP primary (stratum 1) server rackety is a Sun IPC running SunOS 4.1.3 and supporting 734 clients scattered all over the world
- This machine supports NFS, NTP, RIP, IGMP and a mess of printers, radio clocks and an 8-port serial multiplexor
- The mean input packat rate is 6.4 packets/second, which corresponds to a mean poll interval of 157 seconds for each client
- Each input packet generates an average of 0.64 output packets and requires a total of 2.4 ms of CPU time for the input/output transaction
- In total, the NTP service requires 1.54% of the available CPU time and generates 10.5, 608-bit packets per second, or 0.41% of a T1 line
- The conclusion drawn is that even a slow machine can support substantial numbers of clients with no significant degradation on other network services

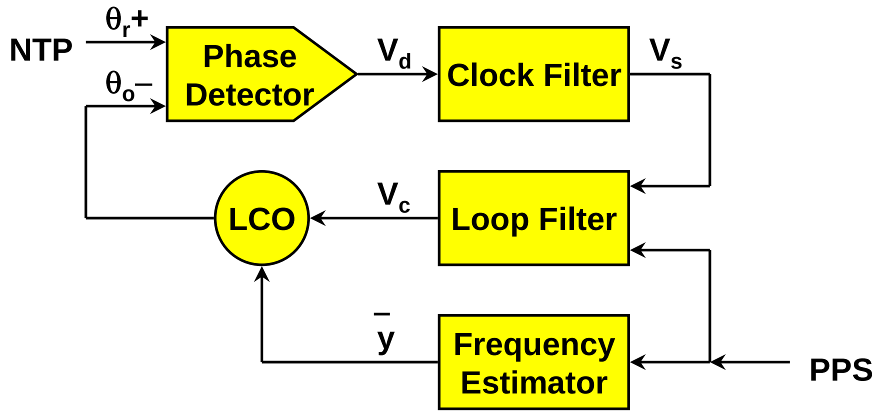
The Sun never sets on NTP

- NTP is argueably the longest running, continuously operating, ubiquitously available protocol in the Internet
- USNO and NIST, as well as equivalents in other countries, provide multiple NTP primary servers directly synchronized to national standard cesium clock ensembles and GPS
- Over 230 Internet primary servers in Australia, Canada, Chile, France, Germany, Isreal, Italy, Holland, Japan, Norway, Sweden, Switzerland, UK, and US
- Over 100,000 Internet secondary servers and clients all over the world
- National and regional service providers BBN, MCI, Sprint, Alternet, etc.
- Agencies and organizations: US Weather Service, US Treasury Service, IRS, PBS, Merrill Lynch, Citicorp, GTE, Sun, DEC, HP, etc.
- Several private networks are reported to have over 10,000 NTP servers and clients behind firewalls; one (GTE) reports in the order of 30,000 NTP-equipped workstations and PCs

NTP enhancements for precision time

- Precision time kernel modifications
  - Time and frequency discipline from NTP or other source
  - Pulse-per-second (PPS) signal interface via modem control lead
  - CPU clock wrangling for symmetric multiprocessor (SMP) systems (Alpha)
- Improved local clock discipline algorithm
  - Hybrid phase/frequency discipline
  - Message intervals extended to several hours for modem services
  - Improved glitch detection and supression
- Precision time and frequency sources
  - PPS signal grooming with median filter and dynamic adaptive time constant
  - Additional drivers for new GPS receivers and PPS discipline
- Reduced hardware and software latencies
  - Serial driver modifications to remove character batching
  - Early timestamp/PPS capture using line disciplines
  - Protocol modifications for multiple primary source mitigation

Improved NTP local clock discipline

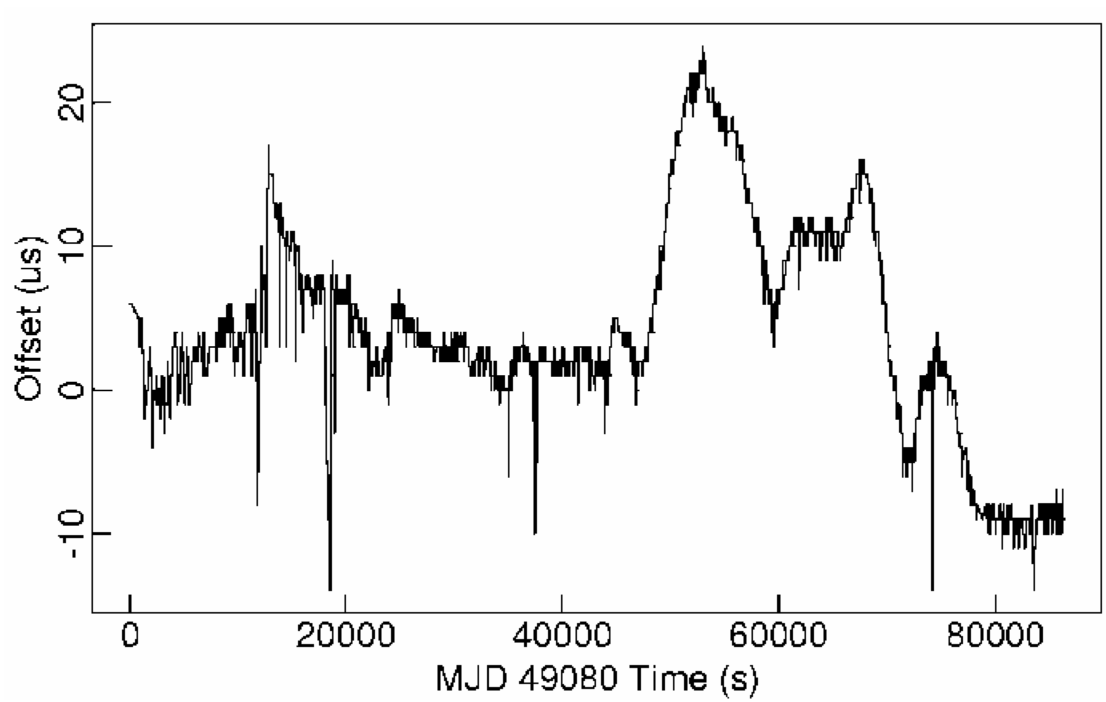


- Type II, adaptive-parameter, hybrid phase/frequency-lock loop estimates LCO phase and frequency
- Phase signal $V_d = \theta_r - \theta_o$ between NTP and local clock oscillator (LCO) computed from timestamps, then filtered to produce control signal V_c
- Auxilliary frequeny-lock loop disciplines LCO frequency *y* to pulse-per-second (PPS) signal, when available
- Loop parameters automatically optimized for update intervals from 16 s to 16384 s

Precision kernel modifications for SMP systems

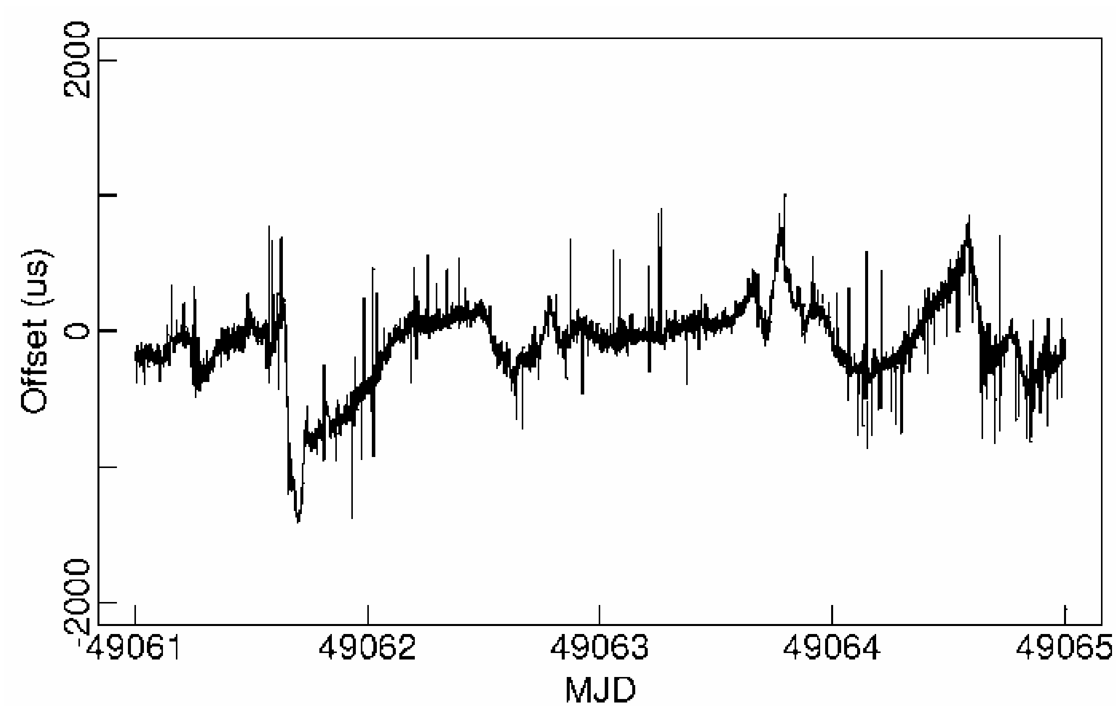
- Each processor has a read-only processor cycle counter (PCC) readable by a special instruction
- The elected master processor increments the kernel clock at each hardware timer interrupt, which occurs at intervals of 1-10 ms
- Once each second, each processor reads the master processor time and estimates the time and frequency offsets of its PCC relative to the kernel clock using an atomic interprocessor interrupt
- When the clock is read on a processor, the current time is computed from the saved master processor time plus an increment computed from the current PCC and estimated time and frequency offsets
- The kernel PLL is adjusted in time and frequency as the result of NTP updates and the PPS signal
- This scheme is now included in Digital Unix 4.0 kernel for the Alpha

PPS signal kernel discipline



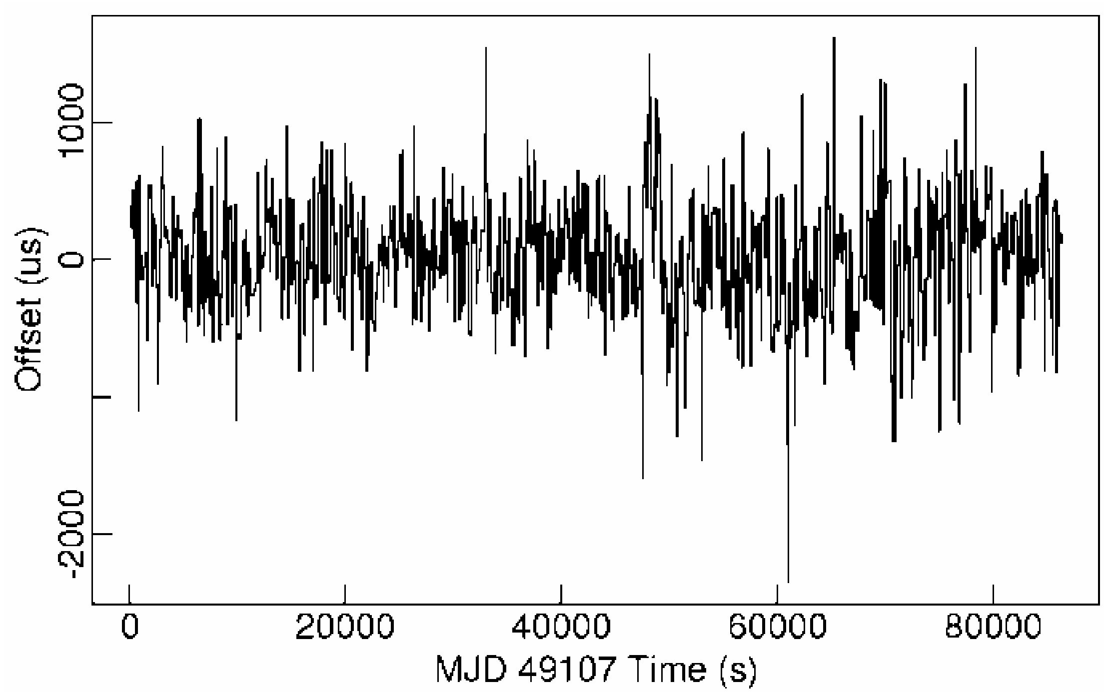
- Graph shows offsets of local clock oscillator relative to PPS signal
- PPS signal disciplines the local clock via kernel interface
- Precision-time kernel modifications are operative

Performance with a secondary server via Ethernet



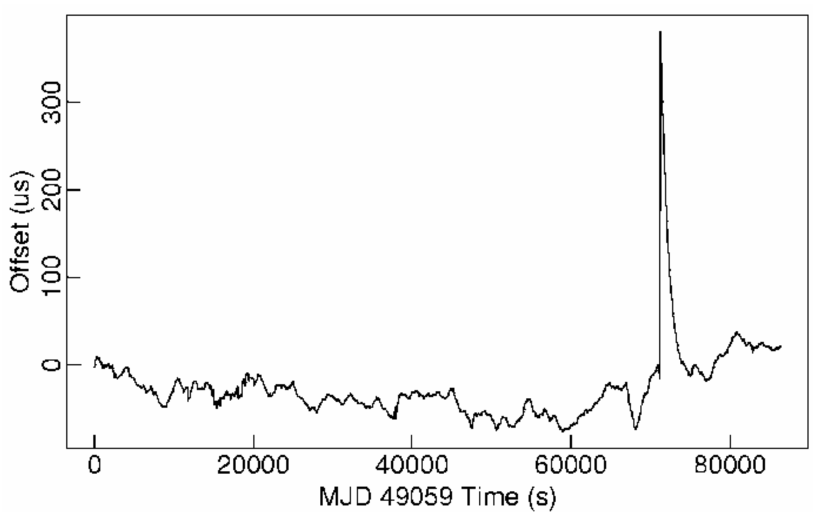
- Clock offsets for Sun SPARC 1+ and SunOS 4.1.1 over four days
  - Primary server synchronized to GPS with PPS
  - Spikes are due to Ethernet jitter and collisions
  - Wander is due to client clock oscillator instability

Performance with a secondary server via T1 line

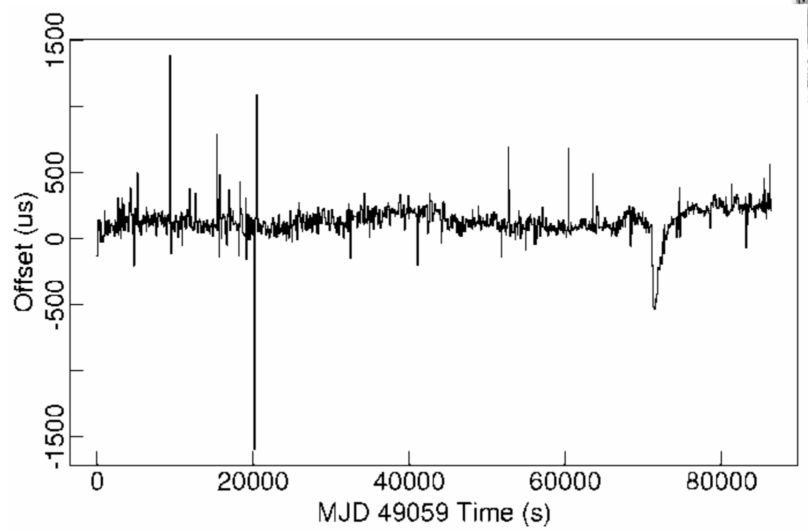


- Clock offsets measured for a NSFnet secondary server running NTP
  - Measurements use NSF server synchronized to a primary server via Ethernets and T1 tail circuit
  - This is typical behavior for lightly loaded T1 circuit

Closed-loop characteristics of primary servers



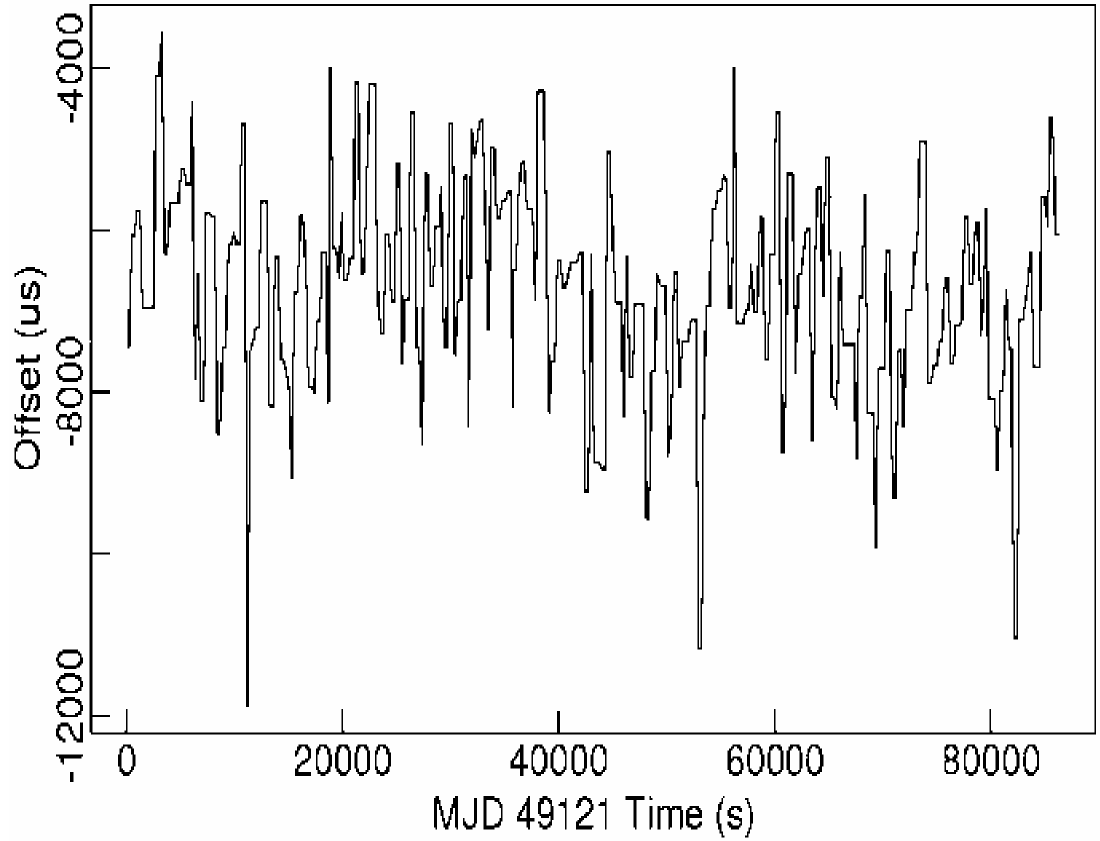
(a) Clock Offset Relative to GPS



(b) Clock Offset between Two Primary Servers

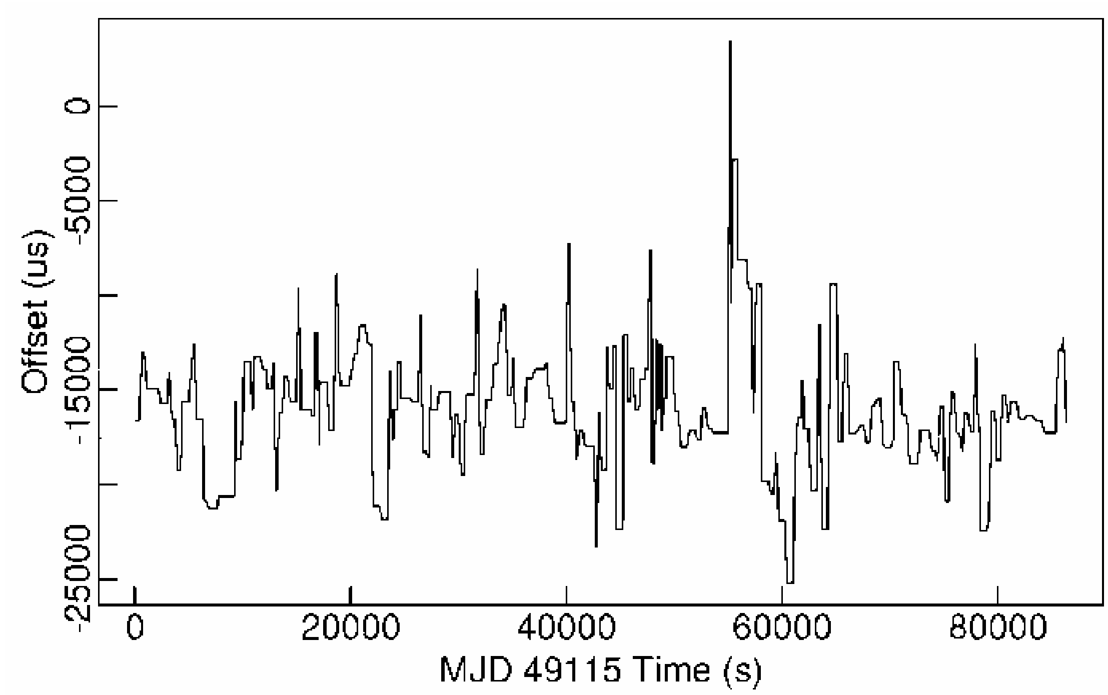
- Clock offsets for Sun SPARC 1+ and SunOS 4.1.1 over one day
  - Two primary servers, both synchronized to the same GPS receiver (no PPS)
  - (a) Measured GPS receiver relative to the local clock of either server
  - (b) Measured one server across the Ethernet relative to the local clock of the other server
  - Note 300-µs spike of unknown cause is visible in both (a) and (b)

Performance with a modem and ACTS service

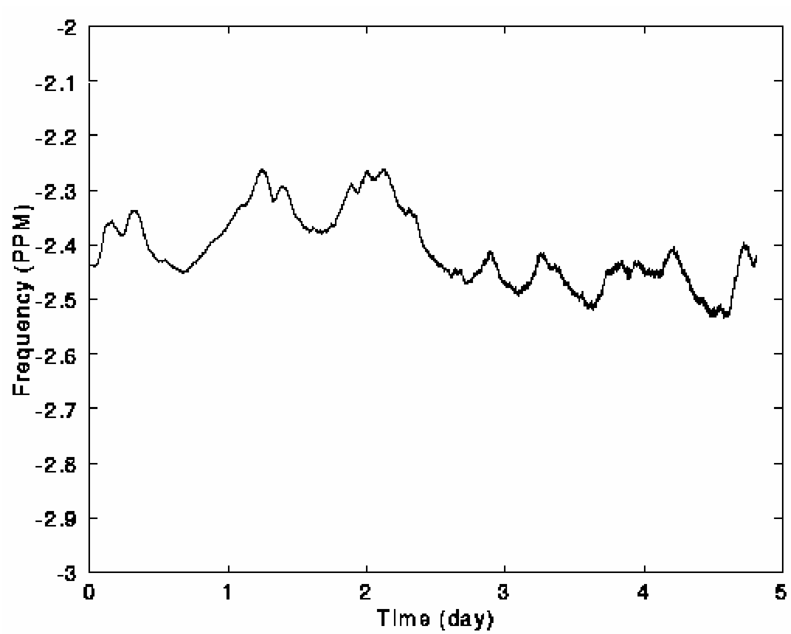


- Measurements use 2300-bps telephone modem and NIST Automated Computer Time Service (ACTS)
- Calls are placed via PSTN at 16,384-s intervals

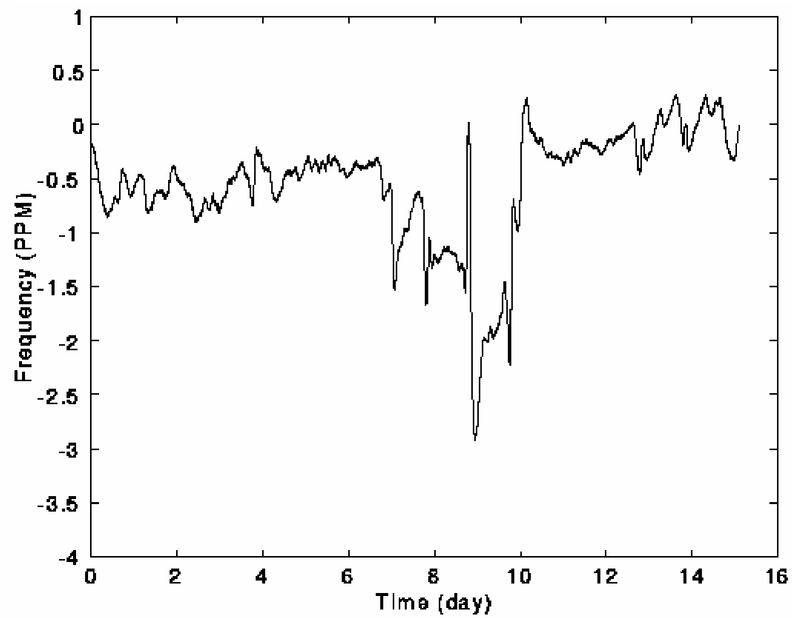
Time offsets with an Australian primary server



Typical frequency variations with temperature



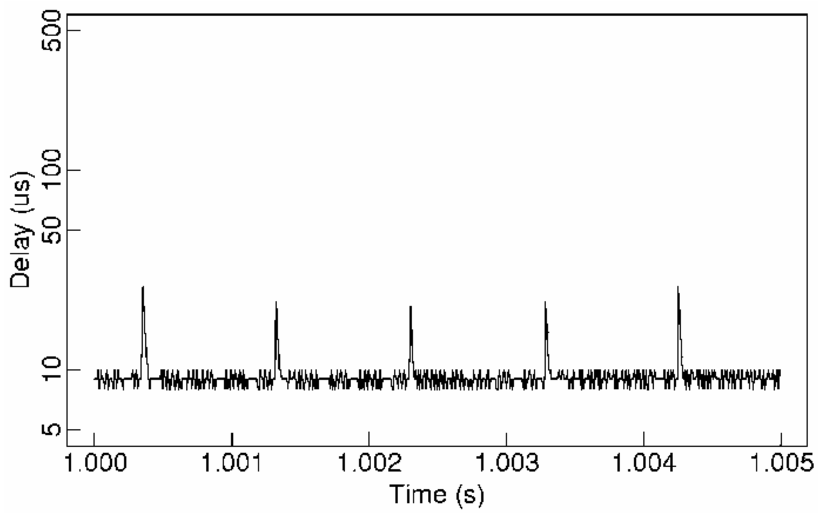
(a) Frequency Offset Measured by PPS



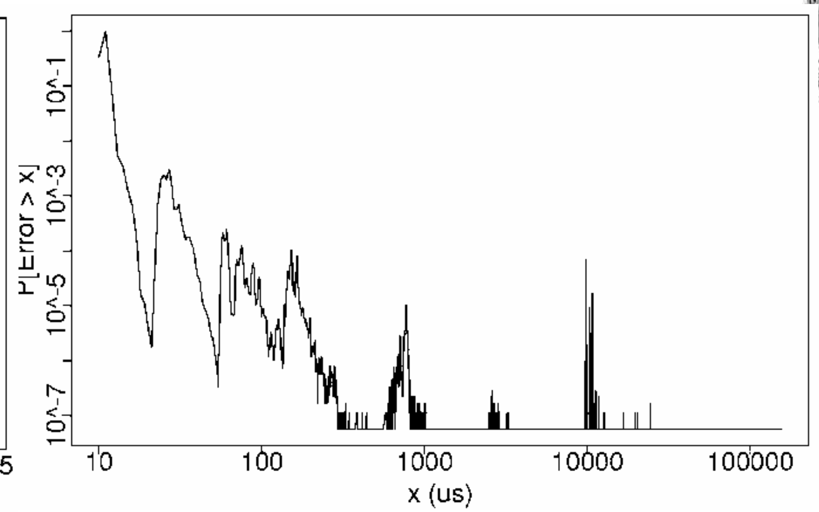
(b) Frequency Offset Measured by NTP

- Measured frequency offsets for free-running local clock oscillator
- (a) Measured directly using PPS signal and `ppsclock` clock discipline
  - Typical room temperature thermostatically controlled in winter
- (b) Measured indirectly using NTP and host synchronized to PPS signal
  - Room temperature follows the ambient in first nice days in spring

Errors due to kernel latencies



(a) Latency for `getimeofday()` Call



(b) Latency Distribution for (a)

- These graphs were constructed using a Digital Alpha and OSF/1 V3.2 with precision time kernel modifications (now standard)
- (a) Measured latency for gettimeofday() call
  - spikes are due to timer interrupt routine
- (b) Probability distribution for (a) measured over about ten minutes
  - Note peaks near 1 ms due timer interrupt routine, others may be due to cache reloads, context switches and time slicing
  - Biggest surprise is very long tail to large fractions of a second

NTP Version 4 current progress and status

- Preliminary NTP Version 4 architecture and algorithms defined
  - Simple NTP (SNTP) Version 4 specification now an Internet draft
  - Documentation completely redone in HTML for browsing
  - Improved local clock model now standard NTP feature
  - Symmetric multiprocessor kernel modifications now in Digital Unix 4.0
- Autonomous configuration
  - Multicast server discovery now standard NTP feature
  - Manycast server discovery implemented and now being tested
  - Distributed add/drop greedy heuristic designed and simulated
  - Span-limited, hierarchical multicast groups using NTP distributed mode and add/drop heuristics under study
- Cryptographic authentication
  - Hybrid symmetric-key/public-key authentication schemes under study
  - Three schemes based on public and private key cryptosystems designed and now being implemented

Future Plans

- Complete NTP Version 4 protocol model, state machine and supporting algorithms
  - Complete design of autonomous configuration and authentication schemes
  - Implement as extensions to existing NTP Version 3 distribution
- Implement, deploy, test and evaluate NTP Version 4 daemon
  - Deploy and test in DARTnet/CAIRN testbed
  - Deploy and test at friendly sites in the US, Europe and Asia
- Prosecute standards agendae in IETF, ANSI, ITU, POSIX
  - Revise the NTP formal specification and launch on standards track
  - Participate in deployment strategies with NIST, USNO, others
- Develop scenarios for other applications such as web caching, DNS servers and other multicast services

NTP online resources

- Internet (Draft) Standard RFC-1305 Version 3
  - Simple NTP (SNTP) Version 3 specification RFC-1769
  - Designated SAFEnet standard (Navy)
  - Under consideration in ANSI, ITU, POSIX
- NTP web page http://www.eecis.udel.edu/~ntp
  - NTP Version 3 release notes and HTML documentation
  - List of public NTP time servers (primary and secondary)
  - NTP newsgroup and FAQ compendium
  - Tutorials, hints and bibliographies
- NTP Version 3 implementation and documentation for Unix, VMS and Windows
  - Ported to over two dozen architectures and operating systems
  - Utility programs for remote monitoring, control and performance evaluation