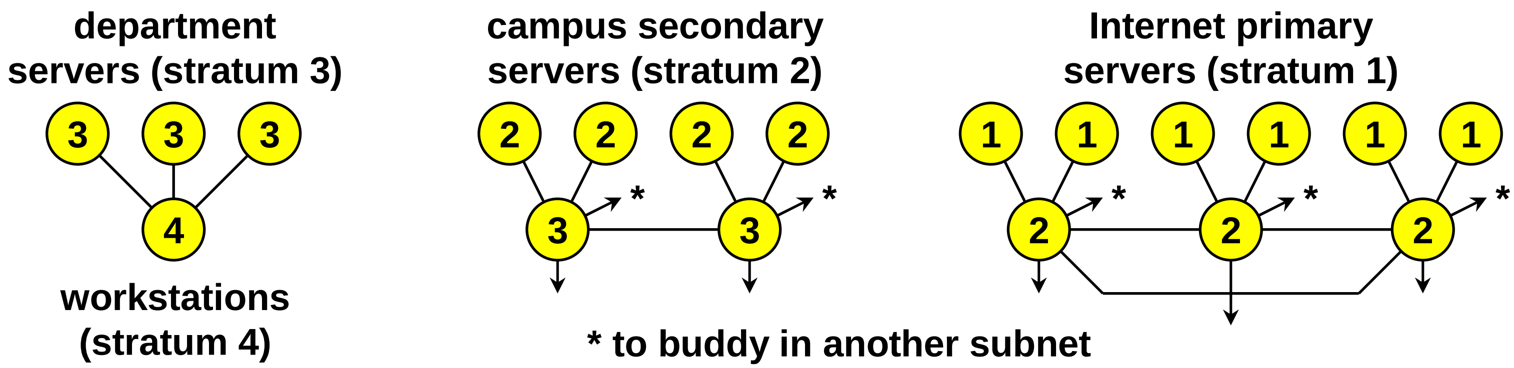
NTP Architecture, Protocol and Algorithms

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

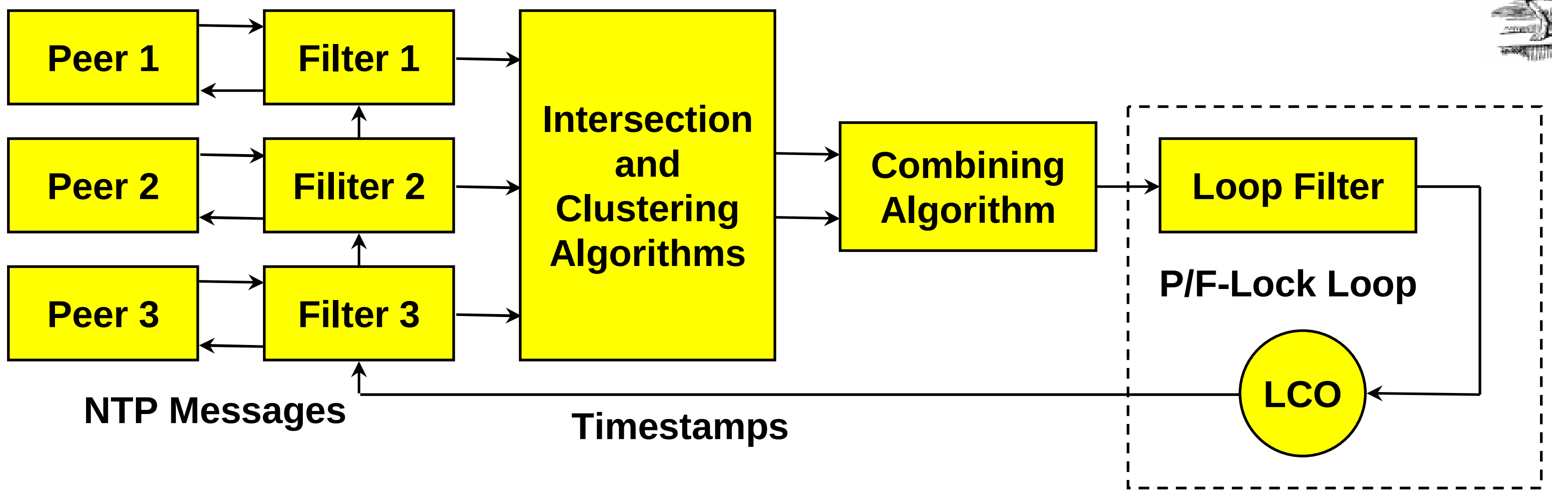
David L. Mills
University of Delaware
http://www.eecis.udel.edu/~mills
mills@udel.edu

The NTP subnet



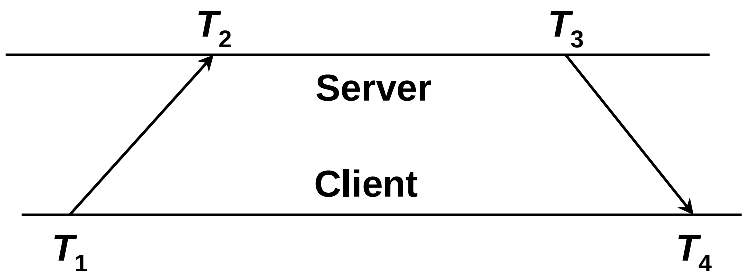
- NTP synchronizes the clocks of hosts and routers in the Internet
- Time synchronization flows from primary servers synchronized via radio and satellite over hierarchical subnet to other servers and clients
- NTP provides submillisecond accuracy on LANs, low tens of milliseconds on typical WANs spanning the country
- NTP software daemon has been ported to almost every workstation and server platform available today, including Unix, Windows and VMS
- Well over 100,000 NTP clients and servers are now deployed in the Internet and its tributaries all over the world

How NTP works



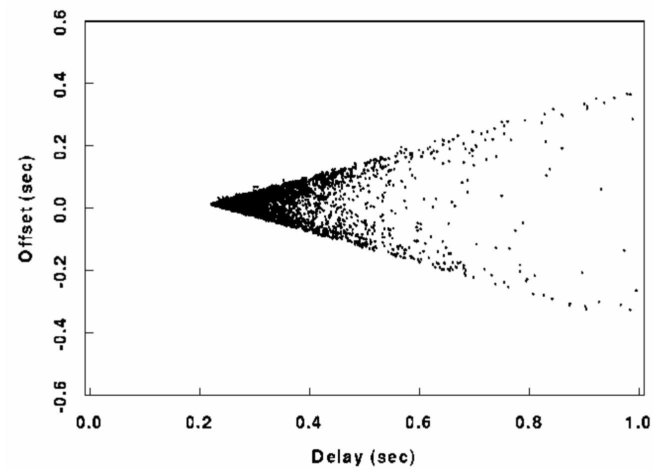
- Multiple servers/peers provide redundancy and diversity
- Clock filters select best from a window of eight clock offset samples
- Intersection and clustering algorithms pick best subset of peers and discard outlyers
- Combining algorithm computes weighted average of offsets for best accuracy
- Loop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

Clock filter algorithm



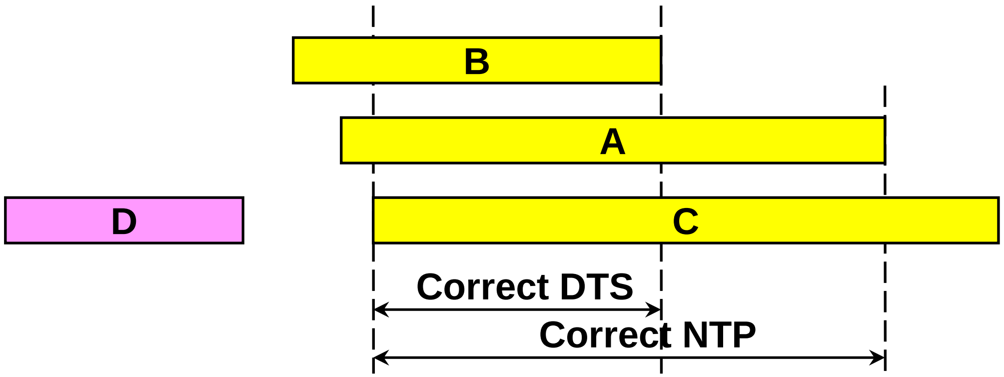
Offset $\theta = \frac{1}{2}[(T_2 - T_1) + (T_3 - T_4)]$

Delay $\delta = (T_4 - T_1) - (T_3 - T_2)$



- Most accurate clock offset θ is measured at the lowest delay δ (apex of the wedge diagram)
- Phase dispersion ε_r is weighted average of offset differences over last eight samples - used as error estimator
- Frequency disperion ε_f represents clock reading and frequency tolerance errors - used in distance metric
- Synchronization distance $\lambda = \varepsilon_f + \delta/2$ - used as distance metric and maximum error bound, since correct time θ_0 must be in the range $\theta - \lambda \leq \theta_0 \leq \theta + \lambda$

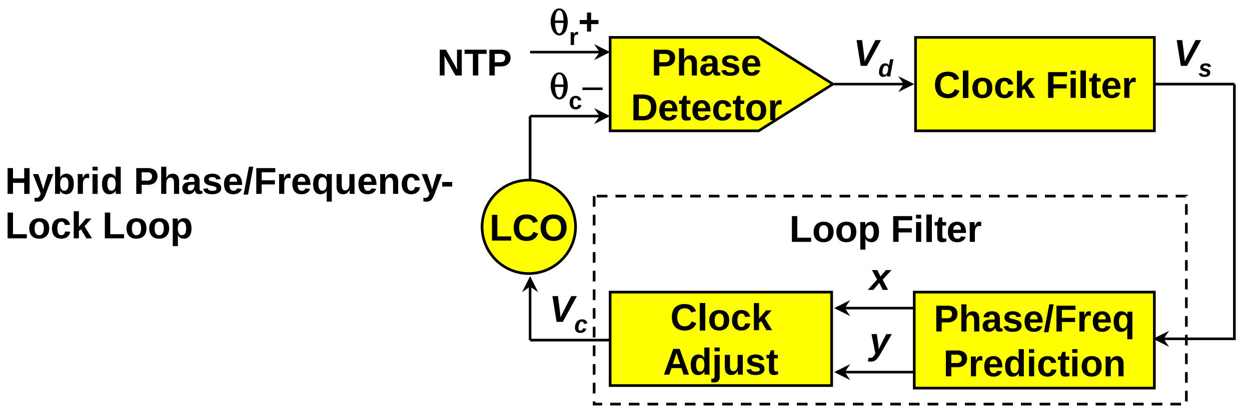
Intersection algorithm



correctness interval = $\theta - \lambda \le \theta_0 \le \theta + \lambda$
m = number of clocks
f = number of presumed falsetickers
A, B, C are truechimers
D is falseticker

- DTS correctness interval is the intersection which contains points from the largest number of correctness intervals
- NTP algorithm requires the midpoint of the intervals to be in the intersection
  - Initially, set falsetickers *f* and counters *c* and *d* to zero
    - Scan from far left endpoint: add one to *c* for every lower endpoint, subtract one for every upper endpoint, add one to *d* for every midpoint
    - If $c \ge m - f$ and $d \ge m - f$, declare success and exit procedure
  - Do the same starting from the far right endpoint
    - If success undeclared, increase *f* by one and try all over again
    - if $f \le m/2$, declare failure

Clock discipline algorithm



- V_d is a function of the phase difference between NTP and LCO
- V_s depends on the stage chosen on the clock filter shift register
- x and y are the phase update and frequency update, respectively, computed by the prediction functions
- Clock adjust process runs once per second to compute V_c, which controls the frequency of the local clock oscillator
- LCO phase is compared to NTP phase to close the feedback loop

Network Time Protocol Security Model and Authentication Scheme

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

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

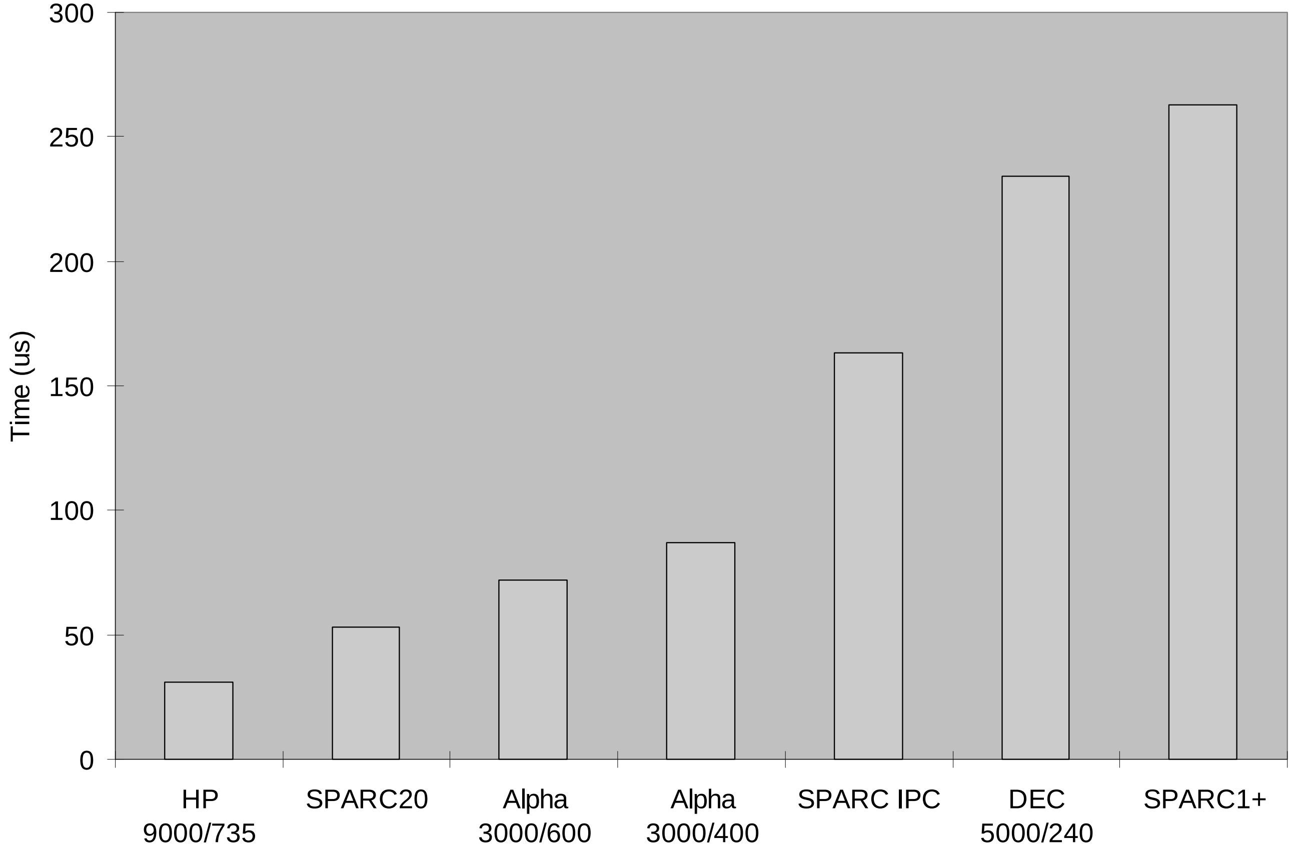
NTP autonomous system model

- Fire-and-forget software
  - Single software distribution can be compiled and installed automatically on most host architectures and operating systems
  - Run-time configuration can be automatically determined and maintained in response to changing network topology and server availability
- Autonomous configuration (autoconfigure)
  - Survey nearby network environment to construct a list of suitable servers
  - Select best servers from among the list using a defined metric
  - Reconfigure the NTP subnet for best accuracy with overhead constraints
  - Periodically refresh the list in order to adapt to changing topology
- Autonomous authentication (autokey)
  - For each new server found, fetch its cryptographic credentials from public databases
  - Authenticate each NTP message received as sent by that server and no other
  - Regenerate keys in a timely manner to avoid compromise

Implementation issues

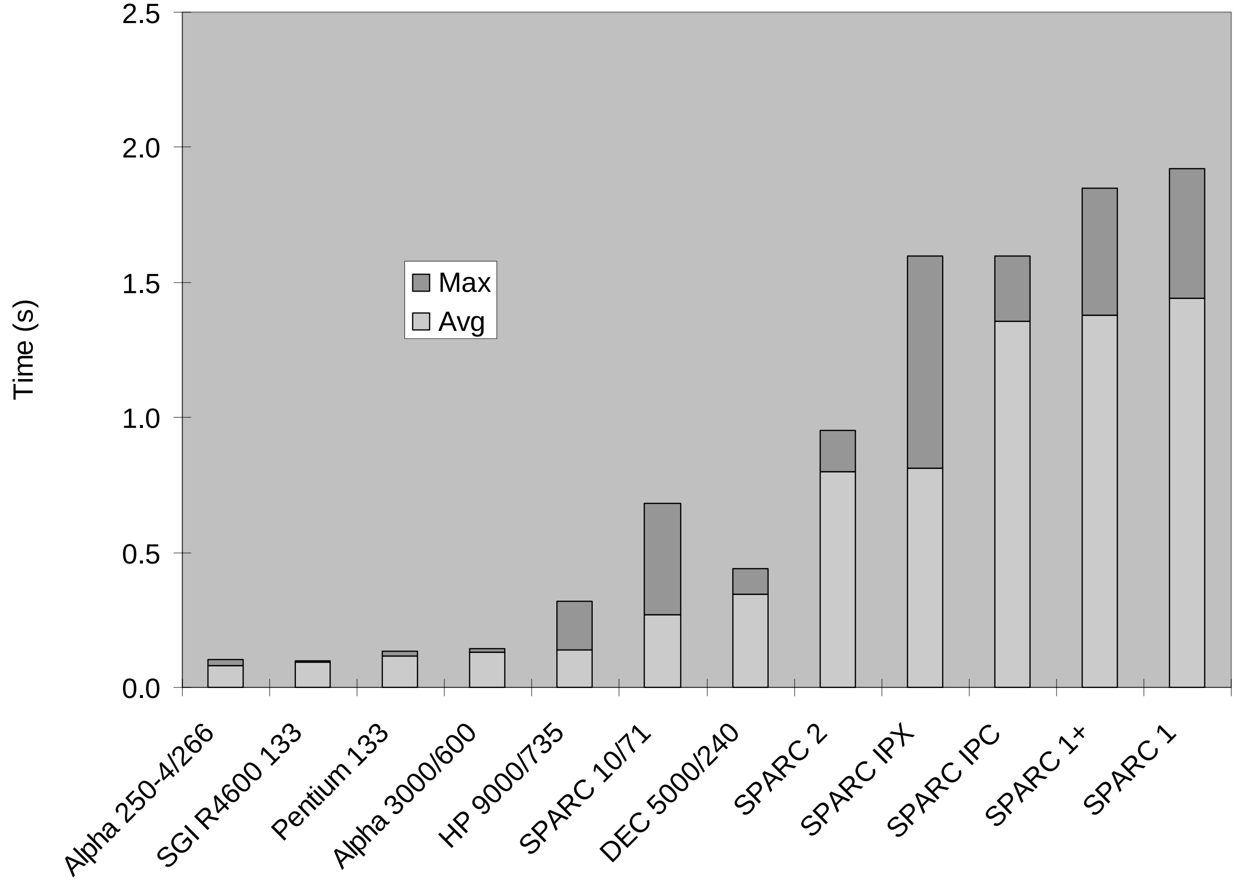
- Public-key cryptography
  - Encryption/decryption algorithms are relatively slow with highly variable running times depending on key and data
  - All keys are random; private keys are never divulged
  - Certificate scheme reliably binds server identification and public key
  - Well suited to multicast paradigm
- Symmetric-key cryptography
  - Encryption/decryption algorithms are relatively fast with constant running times independent of key and data
  - Fixed private keys must be distributed in advance
  - Key agreement (Diffie-Hellman) is required for private random keys
  - Per-association state must be maintained for all clients
  - Not well suited to multicast paradigm

MD5 message digest



- Measured times to construct 128-bit hash of 48-octet NTP header using MD5 algortihm in RSAREF

MD5/RSA digital signature



- Measured times (s) to construct digital signature using RSAREF
- Message authentication code constructed from 48-octet NTP header hashed with MD5, then encrypted with RSA 512-bit private key

NTP authentication - approach

- Authentication and synchronization protocols work independently for each peer, with tentative outcomes confirmed only after both succeed
- Public keys and certificates are obtained and verified relatively infrequently using Secure DNS or equivalent
- Session keys are derived from public keys using fast algorithms
- Each NTP message is individually authenticated using session key and message digest (keyed MD5 or DES-CBC)
- NTP is run individually in unauthenticated mode for each peer to compute offset from system clock, together with related clock data
- If authentication data incomplete, clock data are marked tentative
- If the clock data incomplete, authentication data are marked tentative
- When both authentication and clock data are complete, the peer is admitted to the population used to synchronize the system clock

New extension fields

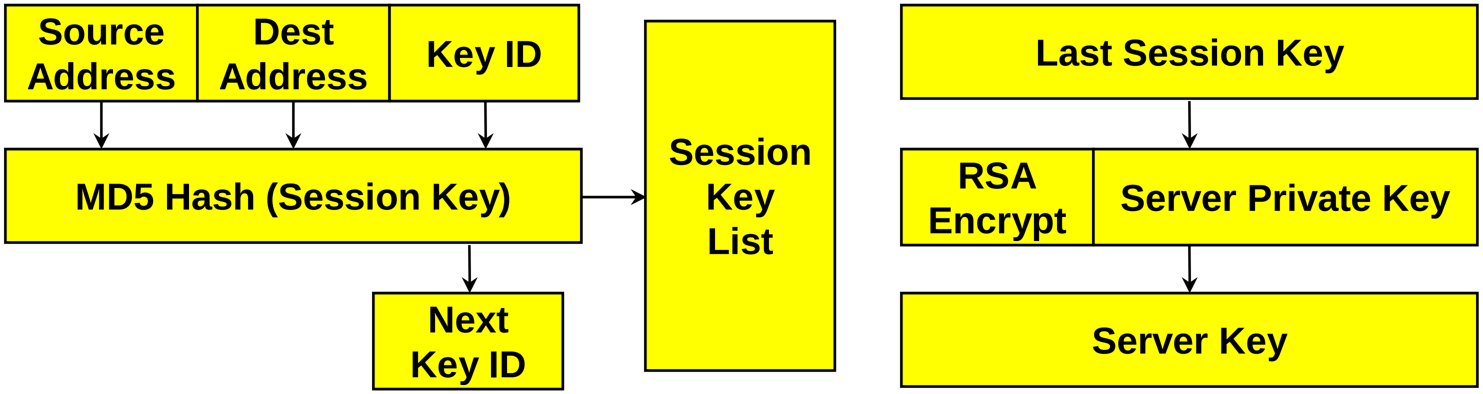
NTP Protocol Header Format (32 bits)

Field Length	Field Type
Sequence Number	
Server Key	

Autokey Extension Field

- New extension field format defined for NTP Version 4 (optional)
  - Fields may be in any order
  - All fields except the last are padded to a 32-bit boundary
  - Last field is padded to a 64-bit boundary
  - Field length covers all payload, including length field, but not padding
- Field types
  - Null/padding - for testing, etc.
  - Certificate - as obtained from directory services (optional)
  - Autokey - in the above format
  - Others as necessary

Generating the session key list



- Server rolls a random 32-bit seed as the initial key ID
- Server generates each session key as hash of IP addresses and key ID
- Low order 32 bits of the session key become the key ID for the next session key
- Server encrypts the last key using RSA and its private key to produce the server key
- Server uses the session key list in reverse order and generates a new one when exhausted

Network Time Protocol Autonomous Configuration

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

Goals and non-goals

- Goals
  - Robustness to many and varied kinds of failures, including Byzantine, fail-stop, malicious attacks and implementation bugs
  - Maximum utilization of Internet multicast services and protocols
  - Depend only on public values and certificates stored in secure directory services
  - Fast operation using a combination of public-key and private-key cryptography
- Non-goals
  - Administrative restrictions (multicast group membership control)
  - Access control - this is provided by firewalls and address filtering
  - Privacy - all protocol values, including time values, are public
  - Protection against out of order or duplicated messages - this is provided by the NTP protocol
  - Non-repudiation - this can be provided by a layered protocol if necessary

Autonomous configuration - approach

- Dynamic peer discovery schemes
  - Primary discovery vehicle using NTP multicast and anycast modes
  - Augmented by DNS, web and service location protocols
  - Augmented by NTP subnet search using standard monitoring facilities
- Automatic optimal configuration
  - Distance metric designed to maximize accuracy and reliability
  - Constraints due to resource limitations and maximum distance
  - Complexity issues require intelligent heuristic
- Candidate optimization algorithms
  - Multicast with or without initial propagation delay calibration
  - Anycast mode with administratively and/or TTL delimited scope
  - Distributed, hierarchical, greedy add/drop heuristic
- Proof of concept based on simulation and implementation with NTP Version 4

NTP configuration scheme

- Multicast scheme (moderate accuracy)
  - Servers flood local area with periodic multicast response messages
  - Clients use client/server unicast mode on initial contact to measure propagation delay, then continue in listen-only mode
- Manycast scheme (highest accuracy)
  - Initially, clients flood local area with a multicast request message
  - Servers respond with multicast response messages
  - Clients continue with servers as if in ordinary configured unicast client/server mode
- Both schemes require effective implosion/explosion controls
  - Expanding-ring search used with TTL and administrative scope
  - Excess network traffic avoided using multicast responses and rumor diffusion
  - Excess client/server population controlled using NTP clustering algorithm and timeout garbage collection

Precision Time Synchronization

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

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

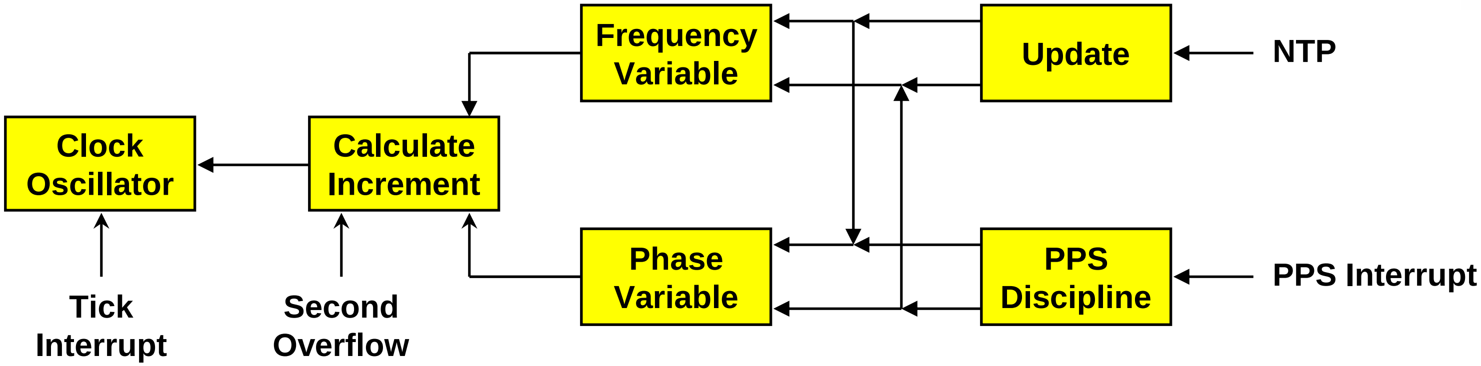
NTP enhancements for precision time

- Reduced hardware and software latencies
  - Serial driver modifications
  - Early timestamp capture in network drivers
- Precision time kernel modifications
  - Time and frequency discipline from NTP or other source
  - Pulse-per-second (PPS) signal interface and user API
- Improved local clock discipline algorithm
  - Time and frequency discipline
  - Reduced impact of jitter and glitches
- Precision time and frequency sources
  - External hardware clock
  - LORAN-C timing receiver
  - WWV/H DSP program for TI 320C25
  - Sun audio codec drivers for IRIG and CHU

Kernel modifications for nanosecond resolution

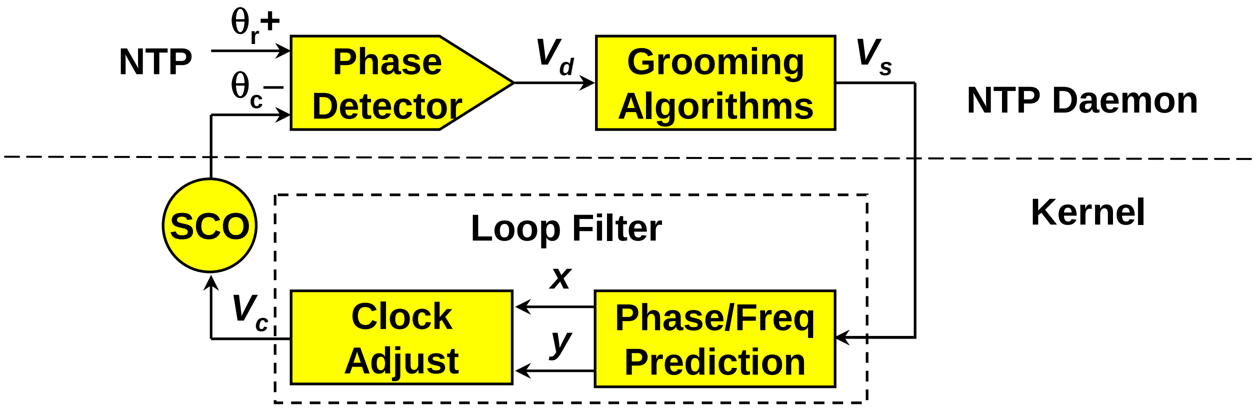
- Package of routines compiled with the operating system kernel
- Represents time in nanoseconds and fraction, frequency in nanoseconds per second and fraction
- Implements nanosecond system clock variable with either microsecond or nanosecond kernel native time variables
- Uses native 64-bit arithmetic for 64-bit architectures, double-precision 32-bit macro package for 32-bit architectures
- Includes two new system calls `ntp_gettime()` and `ntp_adjtime()`
- Includes new system clock read routine with nanosecond interpolation using process cycle counter (PCC)
- Supports run-time tick specification and mode control
- Guaranteed monotonic for single and multiple CPU systems

Nanokernel architecture



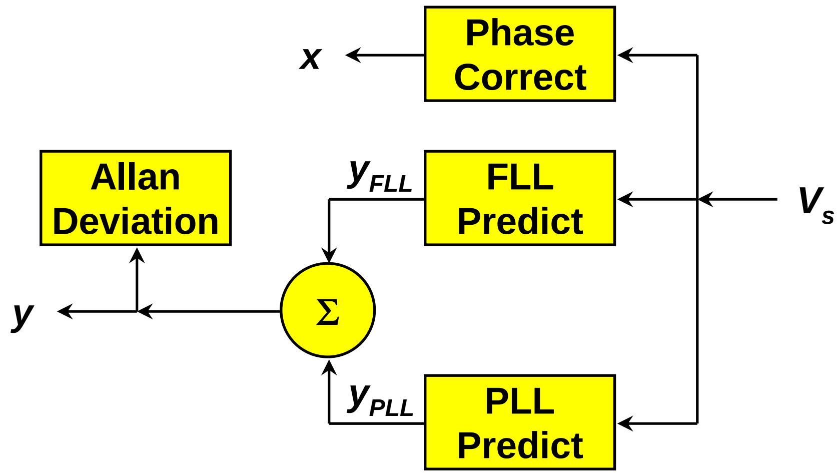
- NTP updates adjust phase and frequency according to time constant at intervals from 64 s to over one day
- On overflow of the clock second, a new increment is calculated for the tick adjustment
- Adjustment is added to system clock at every tick interrupt
- Auxiliary oscillator used to interpret microseconds or nanoseconds between tick interrupts
- PPS discipline adjusts phase at 64-s intervals, frequency at 256-s intervals

Improved NTP kernel clock discipline



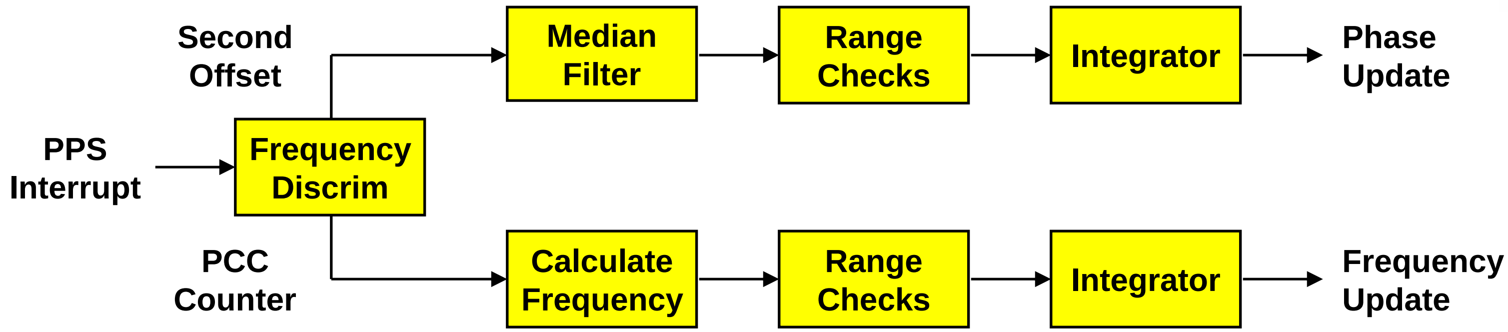
- Type II, adaptive-parameter, hybrid phase/frequency-lock loop estimates system clock oscillator (SCO) phase and frequency
- NTP daemon computes phase error $V_d = \theta_r - \theta_o$ between source and SCO, then grooms samples to produce control signal V_c
- Loop filter computes phase and frequency updates and provides tick adjustments V_c
- SCO adjusted at each hardware tick interrupt

Improved FLL/PLL prediction functions



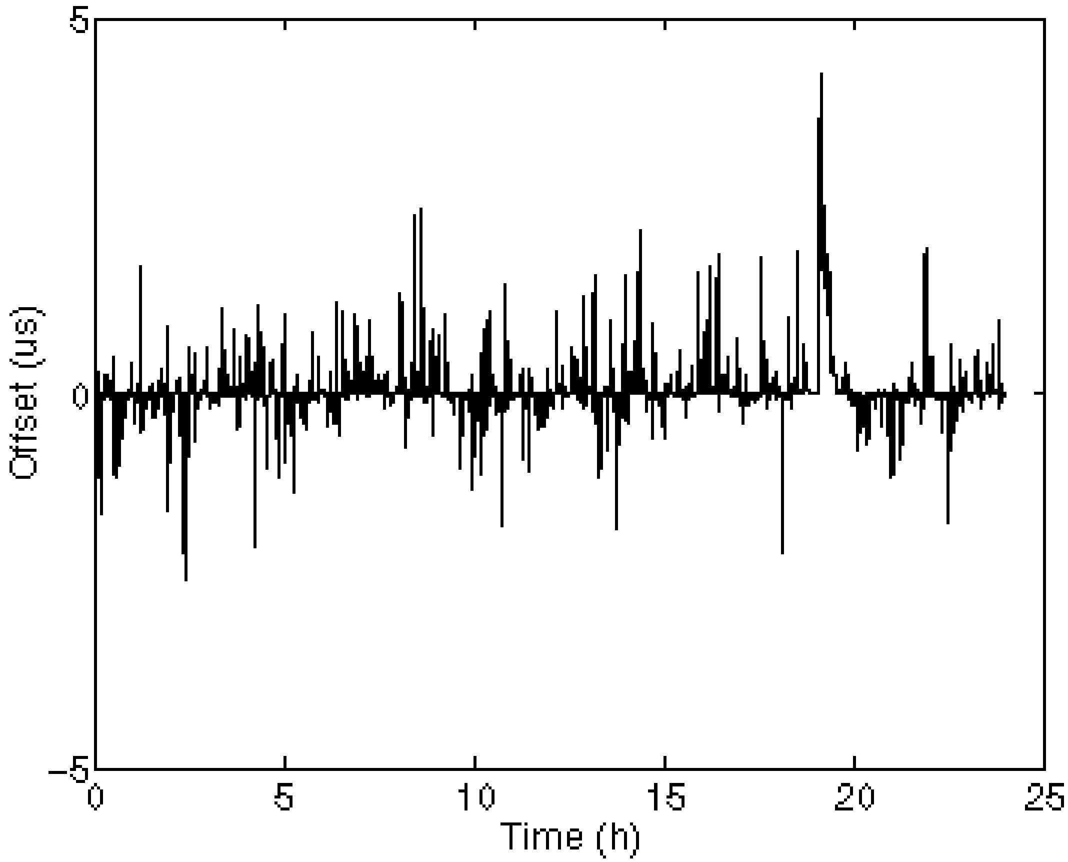
- V_s is the phase offset produced by the data grooming algorithms
- x is the phase correction computed as a fraction of V_s
- y_{FLL} is the frequency adjustment computed as the average of past frequency offsets
- y_{PLL} is the frequency adjustment computed as the integral of past phase offsets
- y_{FLL} and y_{PLL} are combined according to weight factors computed from update interval and Allan deviation predictor

Improved PPS phase and frequency discipline



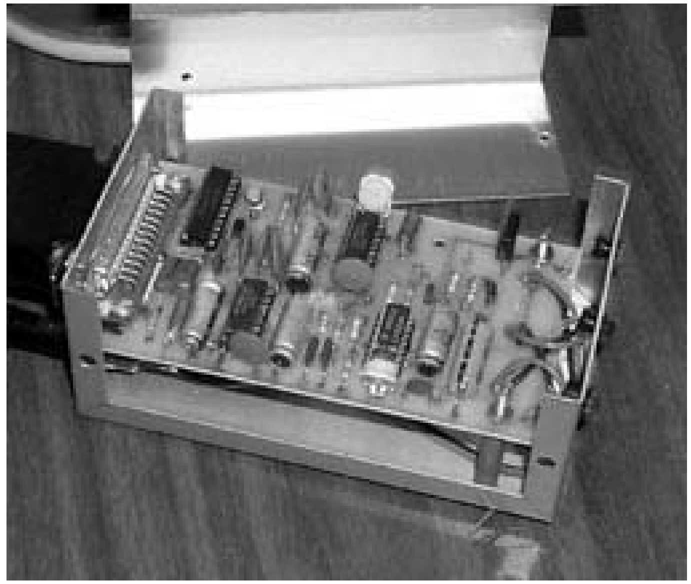
- Phase and frequency disciplined separately - phase from system clock second offset, frequency from process cycle counter (PCC)
- Frequency discriminator rejects noise and misconfigured connections
- Median filter rejects sample outlyers and provides error statistic
- Nonlinear range check filters reject burst errors in phase and frequency
- Phase offsets integrated over 64-s interval
- Frequency offsets integrated over 256-s interval

Residual errors with Digital 433au Alpha



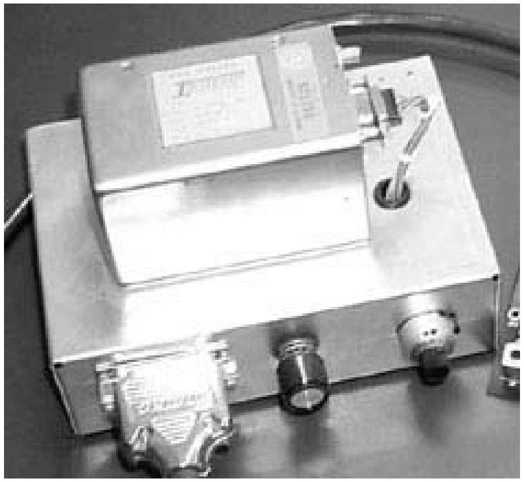
- Graph shows jitter with PPS signal from GPS receiver
- Principal error contribution is due to long unterminated signal cable

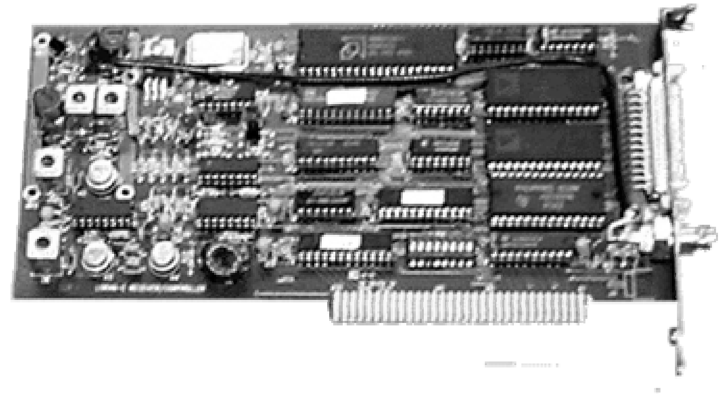
Gadget Box PPS interface



- Used to interface PPS signals from GPS receiver or cesium oscillator
  - Pulse generator and level converter from rising or falling PPS signal edge
  - Simulates serial port character or stimulates modem control lead
- Also used to demodulate timecode broadcast by CHU Canada
  - Narrowband filter, 300-baud modem and level converter
  - The NTP software includes an audio driver that does the same thing

LORAN-C timing receiver





- Inexpensive second-generation bus peripheral for IBM 386-class PC with oven-stabilized external master clock oscillator
  - Includes 100-kHz analog receiver with D/A and A/D converters
  - Functions as precision oscillator with frequency disciplined to selected LORAN-C chain within 200 ns of UTC(LORAN) and 10^{-10} stability
  - PC control program (in portable C) simultaneously tracks up to six stations from the same LORAN-C chain
- Intended to be used with NTP to resolve inherent LORAN-C timing ambiguity

Current progress and status

- NTP Version 4 architecture and algorithms
  - Backwards compatible with earlier versions
  - Improved local clock model implemented and tested
  - Multicast mode with propagation calibration implemented and tested
  - Distributed multicast mode protocol designed and documented
- Autonomous configuration *autoconfigure*
  - Distributed add/drop greedy heuristic designed and simulated
  - Span-limited, hierarchical multicast groups using NTP distributed mode and add/drop heuristics under study
- Autonomous authentication *autokey*
  - Ultimate security based on public-key infrastructure
  - Random keys used only once
  - Automatic key generation and distribution
  - Implemented and under test in NTP Version 4

Future plans

- Complete *autoconfigure* and *autokey* implementation in NTP Version 4
- Deploy, test and evaluate NTP Version 4 daemon in DARTnet II testbed, then at friendly sites in the US, Europe and Asia
- Revise the NTP formal specification and launch on standards track
- Participate in deployment strategies with NIST, USNO, others
- Prosecute standards agendae in IETF, ANSI, ITU, POSIX
- Develop scenarios for other applications such as web caching, DNS servers and other multicast services

NTP online resources

- NTP specification documents
  - Internet (Draft) NTP standard specification RFC-1305
  - Simple NTP (SNTP) RFC-2030
  - NTP Version 4 papers and reports at *www.eecis.udel.edu/~mills*
  - Under consideration in ANSI, ITU, POSIX
- NTP web page *www.eecis.udel.edu/~ntp*
  - NTP Version 3 and Version 4 software and HTML documentation
    - Utility programs for remote monitoring, control and performance evaluation
    - Ported to over two dozen architectures and operating systems
  - Supporting resources
    - List of public NTP time servers (primary and secondary)
    - NTP newsgroup and FAQ compendia
    - Tutorials, hints and bibliographies
    - Links to other NTP software

Further information

- Network Time Protocol (NTP): *www.eecis.udel.edu/~ntp*
  - Current NTP Version 3 and 4 software and documentation
  - FAQ and links to other sources and interesting places
- David L. Mills: *www.eecis.udel.edu/~mills*
  - Papers, reports and memoranda in PostScript and PDF formats
  - Briefings in HTML, PostScript, PowerPoint and PDF formats
  - Collaboration resources hardware, software and documentation
  - Songs, photo galleries and after-dinner speech scripts
- FTP server *ftp.udel.edu* (`pub/ntp` directory)
  - Current NTP Version 3 and 4 software and documentation repository
  - Collaboration resources repository
- Related project descriptions and briefings
  - See “Current Research Project Descriptions and Briefings” at *www.eecis.udel.edu/~mills*