

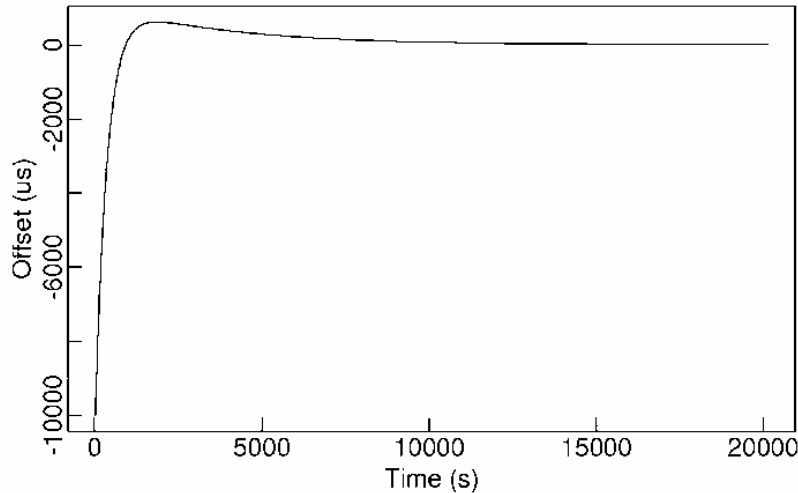
NTP Clock Discipline Principles

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

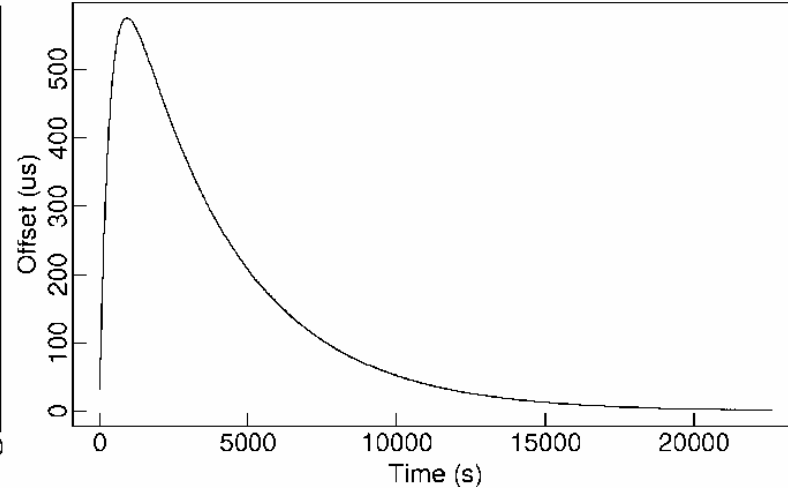


Sir John Tenniel; *Alice's Adventures in Wonderland*, Lewis Carroll

Traditional approach using phase-lock loop (PLL)



Response to 10-ms Phase Step



Response to 2-PPM Frequency Step

- Left graph shows the impulse response for a 10-ms time step and 64-s poll interval using a traditional linear PLL.
- Right graph shows the impulse response for a 5-PPM frequency step and 64-s poll interval.
- It takes too long to converge the loop using linear systems.
- A hybrid linear/nonlinear approach may do much better.

Clock discipline design principles



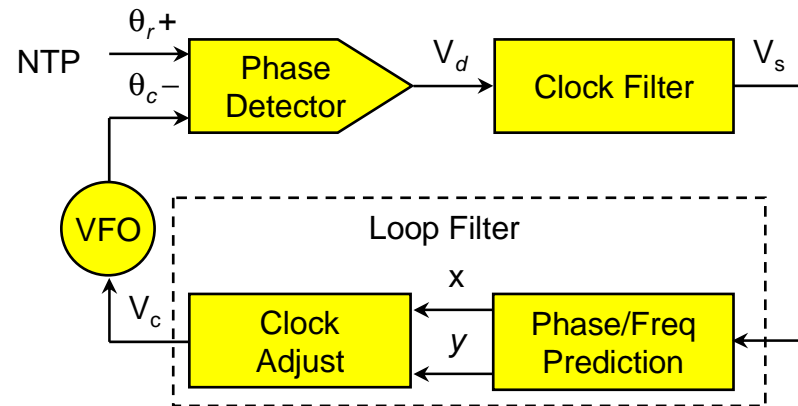
- The clock discipline algorithm functions as a nonlinear, hybrid phase/frequency-lock (NHPFL) feedback loop.
- Detailed computer clock analysis yields the optimum averaging interval depending on prevailing network jitter and oscillator wander.
- Optimum value is determined in real time by measuring the jitter and wander separately.
- Clock state machine quickly converges time and frequency and suppresses transients resulting from leap events, etc.
- Huff&puff algorithm corrects for large outliers and asymmetric delays
- Popcorn spike suppressor clips noise spikes.

Clock discipline design approach



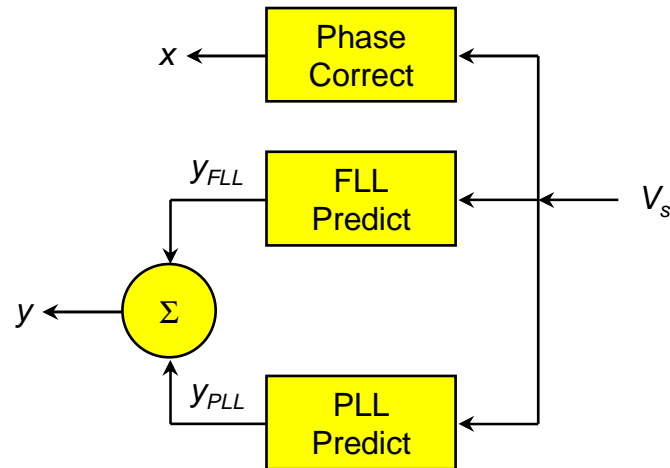
- Phase noise due to network jitter prevails at the lower poll intervals, so a second-order phase-lock loop (PLL) is the best frequency predictor.
- Frequency noise due to random-walk oscillator wander prevails at the higher poll intervals, so a first-order frequency-lock loop (FLL) is the best frequency predictor.
- A crafted heuristic algorithm is necessary to combine both predictions.
- The NHPFL algorithm combines the time and frequency predictions in a seamless way for poll intervals from 16 seconds to 36 hours.
 - The PLL frequency adjustment is computed as the integral of past frequency offsets.
 - The FLL frequency adjustment is computed as the exponential average of past frequency offsets.
 - An additional phase adjustment is necessary for loop stability.
 - The poll interval, which determines the loop time constant, is determined in response to measured jitter and wander.

Clock discipline algorithm



- V_d is a function of the NTP and VFO phase differences.
- V_s depends on the stage chosen of the clock filter shift register.
- x is the phase correction and y the frequency adjustment computed by the prediction functions.
- The clock adjust process runs once each second to adjust the VFO phase by V_c .
- The loop behavior is determined by the loop filter parameters.

FLL/PLL prediction functions



- V_s is the phase offset produced by the clock filter algorithm.
- x is the phase correction computed as the value of V_s .
- y_{FLL} is the frequency prediction computed as the average of past values of V_s .
- y_{PLL} is the frequency prediction computed as the integral of past values of V_s .
- y_{FLL} and y_{PLL} are combined according to weight factors determined by poll interval, update interval and Allan intercept.

Detailed calculations



- The phase correction x and frequency predictions y_{PLL} and y_{FLL} are recalculated at each clock update.

$$y_{PLL} = \frac{\min(\mu, 2^\tau)\theta}{(4K_{PLL}2^\tau)^2} \quad y_{FLL} = \frac{\Delta_\theta}{K_{FLL} \max(\mu, A_x)} \quad x = \theta \quad \begin{matrix} \zeta = 1.2 & 2^{\tau+4} > A_x \\ \zeta = 1 & \text{otherwise} \end{matrix}$$

- The VFO adjustment V_C is updated by the clock adjust process at one-second intervals.

$$dx = \frac{x}{K_{PLL}2^{\zeta\tau}} \quad V_C = y_{PLL} + y_{FLL} + dx \quad x = x - dx$$

- Constants

$K_{PLL} = 16$	PLL gain
$K_{FLL} = 8$	FLL gain
$A_x = 1024s$	Allan intercept

- Variables

τ	poll interval (\log_2)
μ	update interval
θ	clock offset
Δ_θ	offset change since last update
ζ	damping factor

Poll adjust strategy



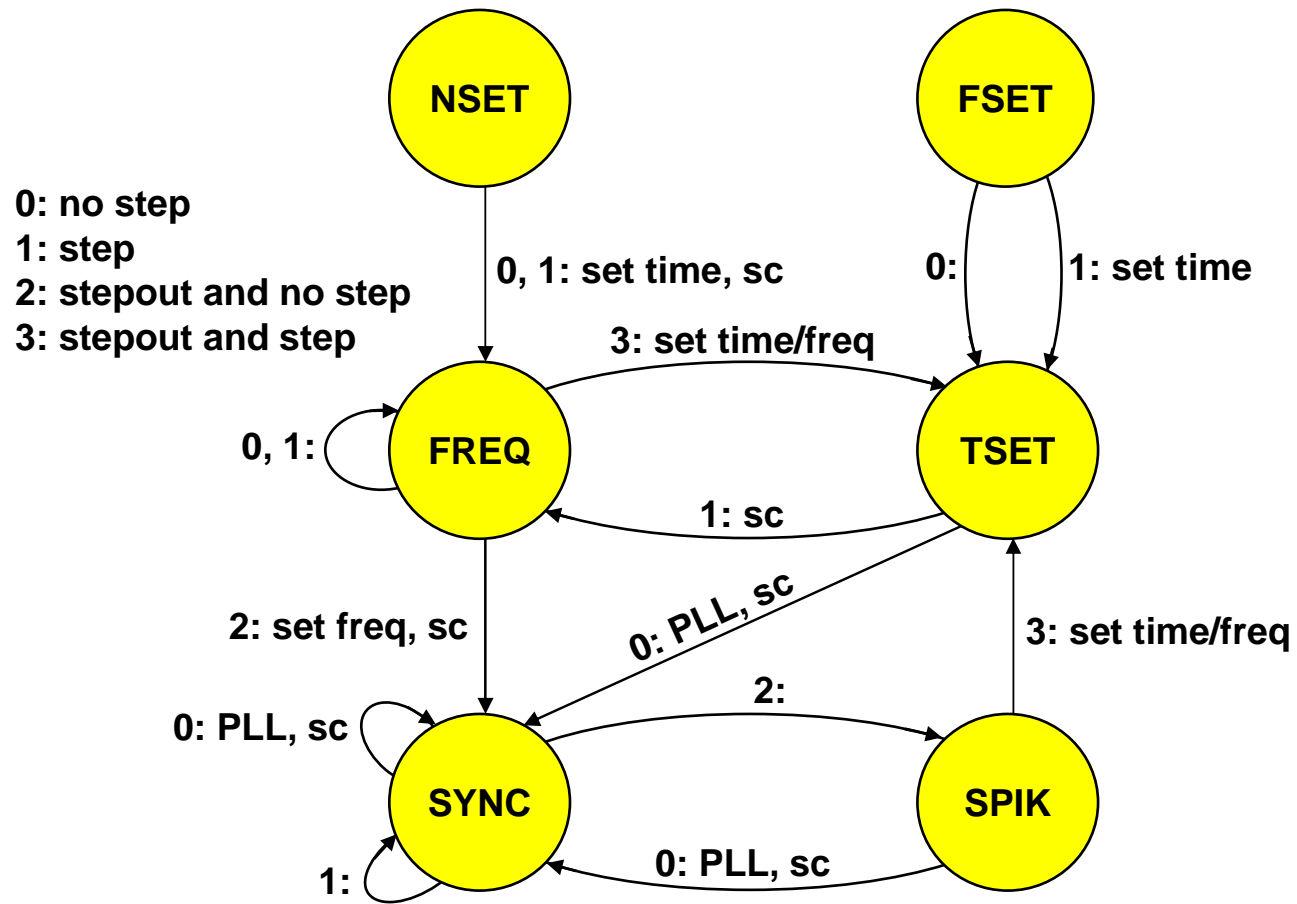
- Note that as τ increases the phase noise $\langle\varphi_P\rangle$ decreases with slope -1 , while the frequency noise $\langle\varphi_F\rangle(\tau \ll 1)$ increases with slope $+0.5$. Thus, the minimum error is when $\langle\varphi_P\rangle = \langle\varphi_F\rangle(\tau \ll 1)$. (Remember that τ is \log_2 of the actual poll interval.) Thus, the strategy is:
- If $\langle\varphi_P\rangle > \langle\varphi_F\rangle(\tau \ll 1)$ and $|\theta| < K_G\langle\varphi_P\rangle$, increase the hysteresis counter h by τ .
 - If $h > K_H$, set $h = 0$ and increase τ by one.
- Else, decrease h by two, in order to adapt to rapid frequency changes.
 - if $h < -K_H$, set $h = 0$ and decrease τ by one.
- Constants
 - $K_H = 30$ hysteresis limit
 - $K_G = 4$ hysteresis threshold
- Variables
 - $\langle\varphi_P\rangle$ average phase differences
 - $\langle\varphi_F\rangle$ average frequency differences
 - h hysteresis counter

State machine operations

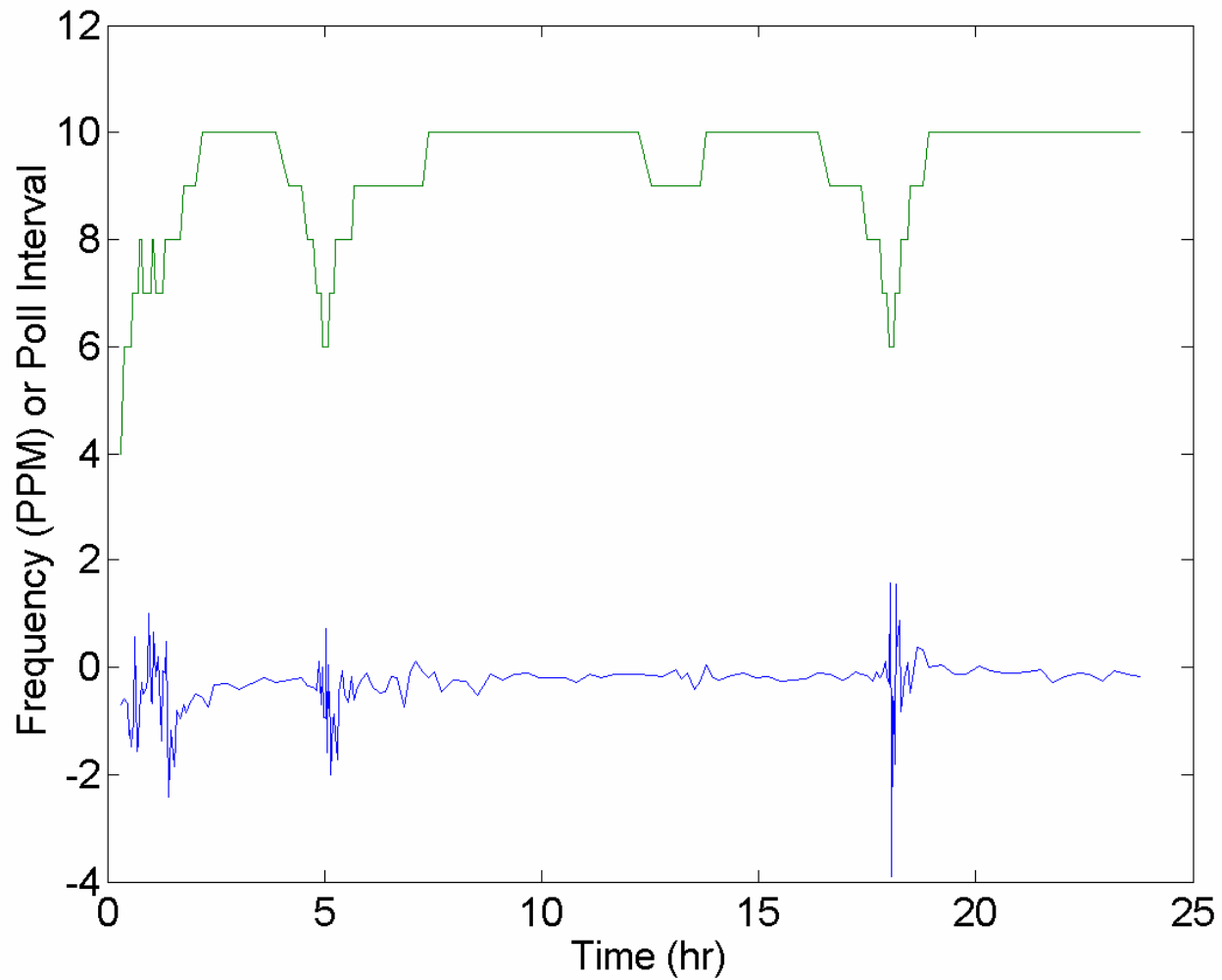


- There are three thresholds which affect the state machine.
 - Panic threshold (1000 s): exit to the operating system if offset exceeds.
 - Step threshold (128 ms): ignore if offset exceeds until stepout.
 - Stepout threshold (900 s): interval within which step spikes are ignored.
- When the discipline is started for the first time, set the time and calculate a possibly large frequency correction.
- Subsequently when the discipline is started, set the time only if the offset exceeds the step threshold.
- When calculating the frequency correction, continue to the stepout threshold in order to produce an accurate value, then set the time and frequency.
- Once the initial time and frequency have been set, run the HNPFL algorithm and the poll-adjust algorithm. Ignore transients greater than the step threshold, unless the stepout threshold is exceeded.

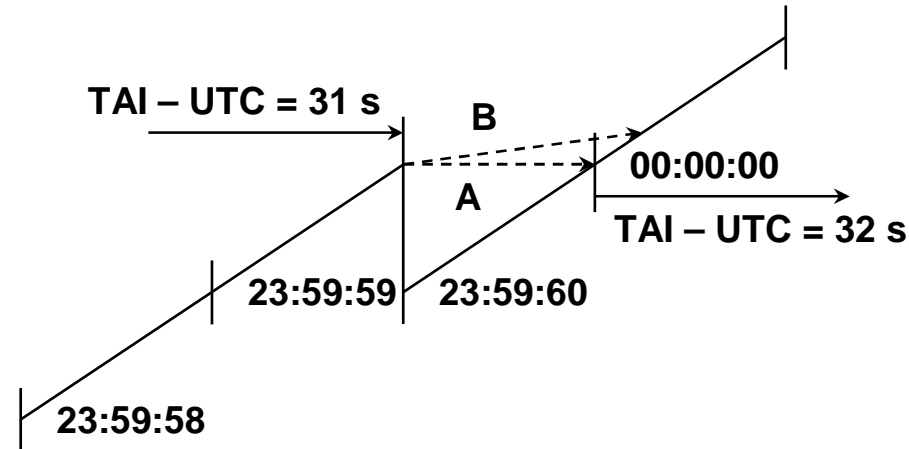
Clock state machine transition function



Frequency offset and poll interval from simulator



Leap second insertion



- Hardware time is read from the processor cycle counter that increments in the low nanosecond range.
- Software time may not step backward; it must increment forward at least 1 ns for every reading.
- The clock is stepped backward at leap second 23:59:59, but software time stays the same (A), unless the clock is read.
- At the end of the leap second 23:59:60 the clock is ahead (B) in nanoseconds the number of times it was read.

Clock discipline algorithm performance



- The algorithm converges time within 5 ms and frequency within 2 PPM in a very short time with poll intervals up to 10 (1024 s).
 - Time to converge with no frequency file is less than 20 min.
 - Time to converge with frequency file and no iburst is less than 4 min.
 - Time to converge with frequency file and iburst is less than 10 s.
 - Previous designs could take days to achieve this performance.
- Following slides show results from a simulator run for typical LAN
 - Initial oscillator frequency offset -400 PPM with wander parameter 1 s/s.
 - Initial time offset 600 s with network jitter parameter 1 ms.
 - These are parameters typical for 10 Mb Ethernets and computer oscillators.
- The poll interval rapidly adapts to frequency changes.
 - The frequency (blue) is in PPM.
 - The poll interval (green) is in $\log_2(s)$ units.
 - It increases slowly if jitter is greater than wander and decreases rapidly otherwise.

Further information



- NTP home page <http://www.ntp.org>
 - Current NTP Version 3 and 4 software and documentation
 - FAQ and links to other sources and interesting places
- David L. Mills home page <http://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
- Udel FTP server: <ftp://ftp.udel.edu/pub/ntp>
 - Current NTP Version software, documentation and support
 - Collaboration resources and junkbox
- Related projects <http://www.eecis.udel.edu/~mills/status.htm>
 - Current research project descriptions and briefings