UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

Dependable Distributed Systems – 5880V/UE

Part 2: Distributed Systems: Physical Clock Synchronization – 2021-10-05

Prof. Dr. Hans P. Reiser | WS 2021/22

UNIVERSITÄT PASSAU

Synchronization of physical clocks

Synchronization of physical clocks

[Sir John Tenniel, http://www.gutenberg.org/dirs/1/1/114/]

Problem of clock synchronization

- There are no completely identical physical clocks
  - Different initialization (constant offset)
  - Different speed (frequency error)
  - Subject to environmental conditions (e.g., component ageing, temperature dependency)

  $\Rightarrow$ Without synchronization errors may continuously grow!
- Common clock for all the nodes of a distributed system (usually) not feasible
- Central reference clocks
  - e.g., radio transmission (WWV, DCF77, GPS)

  with technically limited accuracy

Synchronization of physical clocks

Characterizing a clock synchronization algorithm

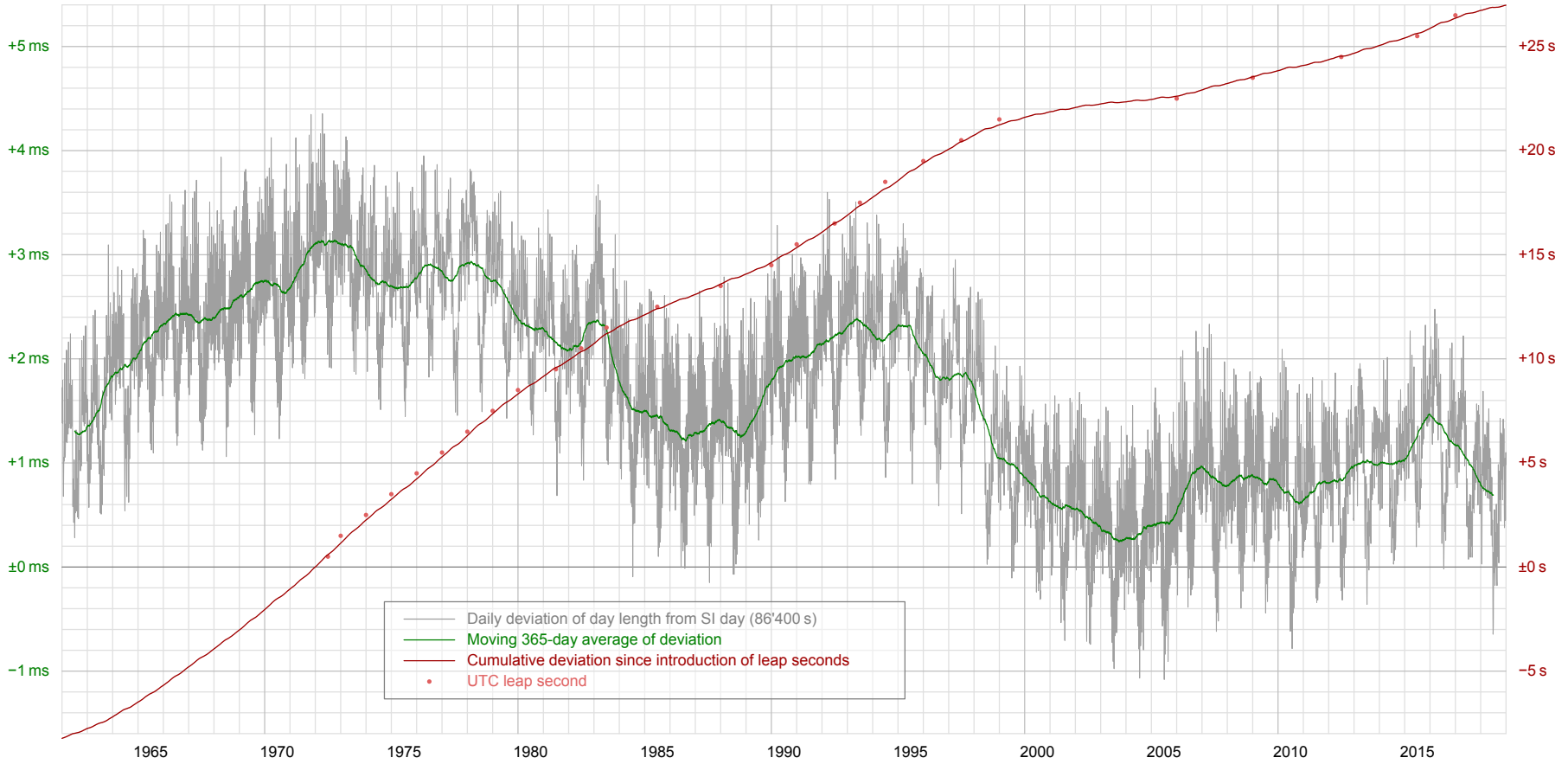
- Clock synchronization can be implemented in multiple ways. . .
- Which properties could be used to characterize an algorithm?

Synchronization of physical clocks

Physical time based on atomic second: TAI

- "The second is 9 192 631 770 times the period of the radiation corresponding to the transition between the two hyperfine levels of the ground state of atoms of the nuclide ^{133}Cs" (International Bureau of Weights and Measures, valid definition since 1967; previously defined by the duration of the Earth's rotation (until 1956) or the duration of a earth circulation of the sun)
- Accuracy of Cs clocks: up to about 10^{-14} (equivalent to $\approx 0.8ns$/day)
- UT1: time defined by astronomical observations
  (mean solar time at the zero meridian)
- UTC (Universal Time Coordinated):
  scale based on atomic second, $|UT1 - UTC| < 0.9s$ using leap seconds (previously 27, last on 31.12.2016)
  Abolishing leap seconds will be discussed on ITU WRC2023 (instead: leap hour in 2600?)

UT1 vs. UTC


+5 ms
+4 ms
+3 ms
+2 ms
+1 ms
±0 ms
−1 ms
+25 s
+20 s
+15 s
+10 s
+5 s
±0 s
−5 s
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
Daily deviation of day length from SI day (86'400 s)
Moving 365-day average of deviation
Cumulative deviation since introduction of leap seconds
UTC leap second

UT1 vs. UTC (pc-magazin.de)

Mail Online

Home | News | U.S. | Sport | TV&Showbiz | Australia | Femail | Health | Science | Money | V

Latest Headlines | News | World News | Arts | Headlines | France | Pictures | Most read | Wires | Discounts

Time hiccup caused online chaos: How the 'leap second' brought down some of the web's most popular sites

By DAMIEN GAYLE

PUBLISHED: 09:49 BST, 2 July 2012 | **UPDATED:** 14:59 BST, 3 July 2012

Share

26 View comments

Some of the web's most popular sites were laid low on Sunday morning after the world's timekeepers added an extra second to the day.

Sites including Reddit, FourSquare, Yelp, LinkedIn, Gawker and

Synchronization of physical clocks

Dissemination of official time: normal frequency / time signal

- Long wave: e.g. DCF77 (Mainflingen, D); WWVB (Boulder, U.S.); maximum accuracy about 50μs
- Short wave: e.g. WWV, WWVH (Hawaii); maximum accuracy about 1ms
- GPS satellites: maximum accuracy around 0.3μs

Synchronization of physical clocks

(DCF77)

(GPS)

Synchronization of physical clocks

Software-based synchronization

- Master-/slave distribution with a reference time (broadcast or poll)
- Distributed consensus

Synchronization of physical clocks

Characterizing a clock synchronization algorithm

- Clock synchronization can be implemented in multiple ways. . .
- Which properties could be used to characterize an algorithm?

Synchronization of physical clocks

Characterization of algorithms:

- Monotony (synchronization causes “jumps” in time?)
- Synchronization with the official time
- Robustness against network partitioning
- Fault tolerance against wrong reference clocks
- Loyalty of reference
- Generated network traffic
- Error estimate

Time in distributed systems

Measuring time offsets

Basic approaches:

- Unidirectional: *A* sends current time to *B*
- Bidirectional: *A* sends request, *B* resonds with current time
- Multiple measurements for increasing accuarcy?

Measuring time offset: unidirectional

Simple:

- A sends its current local time t_A to B
- B records its local time t_B at reception
- Estimate of clock offset: $o = t_A - t_B$
- Error due to message transmission delay, potentially unbounded

Measuring time offset: unidirectional

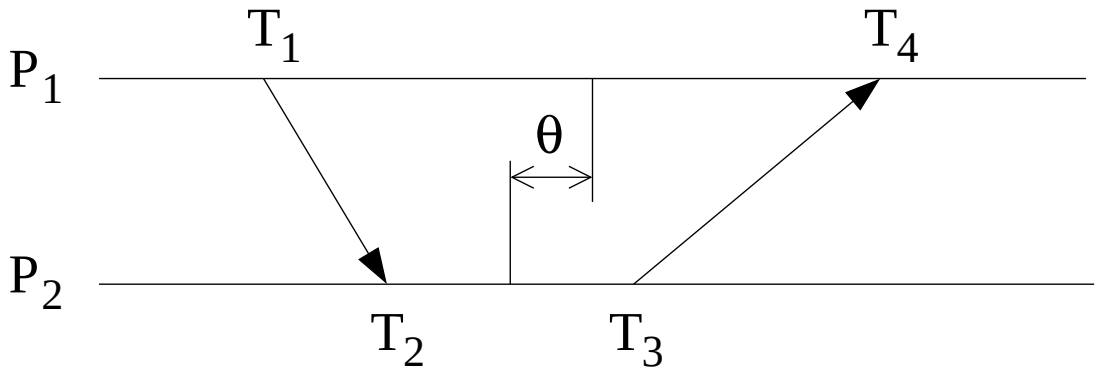
Simple:

- A sends its current local time t_A to B
- B records its local time t_B at reception
- Estimate of clock offset: $o = t_A - t_B$
- Error due to message transmission delay, potentially unbounded

Improved accuracy:

- Estimate message transmission delay d
- Use o to correct offset: $o' = t_A - t_B + d$
- Still errors due to variable message transmission delay and inaccuracy of delay estimation

Measuring time offset: bidirectional



- Total message transmission time: $\delta = (T_4 - T_1) - (T_3 - T_2)$
- Offset estimation: $\theta = (T_2 + T_3)/2 - (T_1 + T_4)/2$
  - Exact value if transmission times equal in both directions
  - Upper bound for error (unbalanced communication): $\delta/2$

Synchronization of physical clocks

Synchronization of physical clocks

Convergence algorithm CNV

- References:
  - L. Lamport, P.M. Melliar-Smith: *Synchronizing Clocks in the Presence of Faults.* Journal of the ACM, VOl. 32, No. 1, 7 (1985) pp. 52-78.

- Tasks of the algorithm:
  - Prevent unbounded growth of divergence between multiple clocks
  - Tolerate faulty clocks

Convergence algorithm CNV

Assumptions

- Initially, all processes have approximately synchronized clocks (error $< \delta$)
- The clocks of all correct processes to run at approximately the correct rate
- A correct process can read the clock of another correct process with a small maximum error ϵ

Goal

- At any time, all correct processes shall have approximately the same local time (error $< \delta$)

Convergence algorithm CNV

Procedure

- Each process reads the clocks of all other processes
- It sets its local clock to the average of all values including its own clock. In the average calculation, it substitutes all values deviating more than δ from its local clock by the value of this local clock
- The desired properties (convergence to a common clock, tolerating faulty clocks) are satisfied if less than one third of the clocks are wrong

Convergence algorithm CNV

Plausibility analysis:

- Definitions
  - p, q: correct processes, r: any process
  - $C_{p,q}$ is the clock of q, as it is known (read) by p
- Observations:
  - r correct $\Rightarrow C_{p,r} \approx C_{q,r}$
  - r faulty $\Rightarrow |C_{p,r} - C_{q,r}| < 3\delta$
- All processes p set their clock to $\sum_i (C_{p,i})/n$

Convergence algorithm CNV

Plausibility analysis:

- Worst case: (n-m) times $(C_{p,i} - C_{q,i}) \approx 0$
  m times $(C_{p,i} - C_{q,i}) < 3\delta$
- So new difference between p and q is $< (3m/n)\delta$
  Assuming $n > 3m$ this means: new difference $< \delta$
- Neglected are:
  - Time for executing the algorithm
  - Error by not reading all clocks exactly simultaneously

Overview

The Network Time Protocol – NTP

Detailed information on NTP on the WWW:
`http://www.ntp.org` (“official NTP website”)
`http://www.eecis.udel.edu/~mills/ntp.html`
(Homepage David Mills)

History:

- Developed since 1982 (NTP v1, RFC 1059) under the direction of David Mils
- Since 1990, NTP v3 (still in use)
- Current Version: NTP v4, since 1994
- Current research by David Mills (University of Delaware): Interplanetary time synchronization for IPIN (Interplanetary Internet, NASA / DARPA project)

The Network Time Protocol – NTP

Properties of NTP

- Purpose: Synchronize computer clocks in the existing Internet
- NTP Pool Project: $>$ 2000 servers worldwide
- Achievable accuracies of about 0.01s in WANs, $<$ 1ms in LANs
- NTP daemon available on almost all computer platforms
- Fault tolerant

The Network Time Protocol – NTP

Basic overview

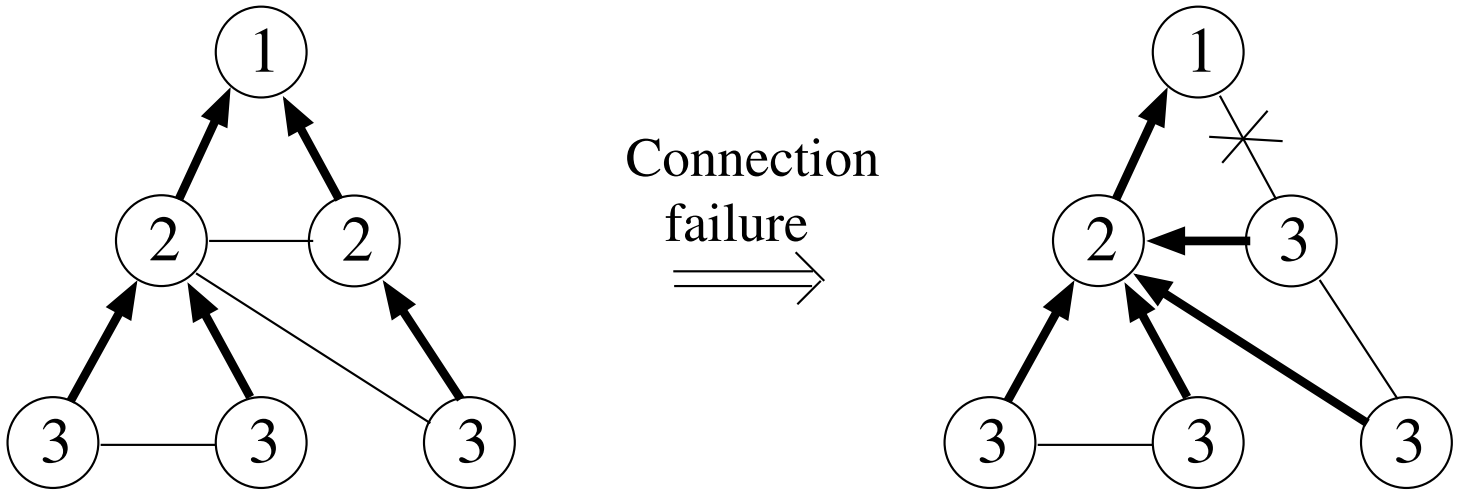
- Primary server (stratum 1), synchronized by radio or leased lines directly to official time standards
- Synchronization of additional nodes through a hierarchical network linked to the primary servers
- Different modes (master/slave, symmetric synchronization, multicast; each with / without authentication)
- Reliability obtained by redundant servers and network paths
- Optimized algorithms in order to reduce errors caused by jitter, changing reference clocks and faulty server
- Local clocks are regulated in time and frequency by an adaptive algorithm

The Network Time Protocol – NTP

Stratum:

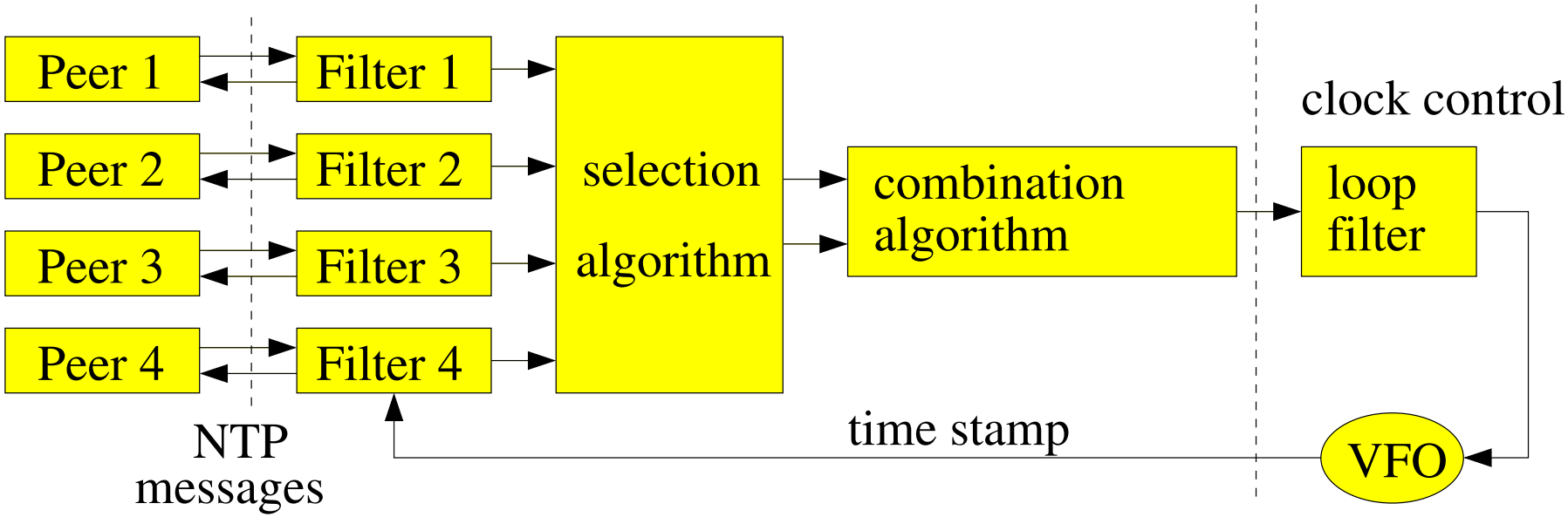
- 1: primary reference clock
- $i > 1$: synchronized with reference clock of stratum $i - 1$

Stratum can change dynamically:



The Network Time Protocol – NTP

Architecture overview:

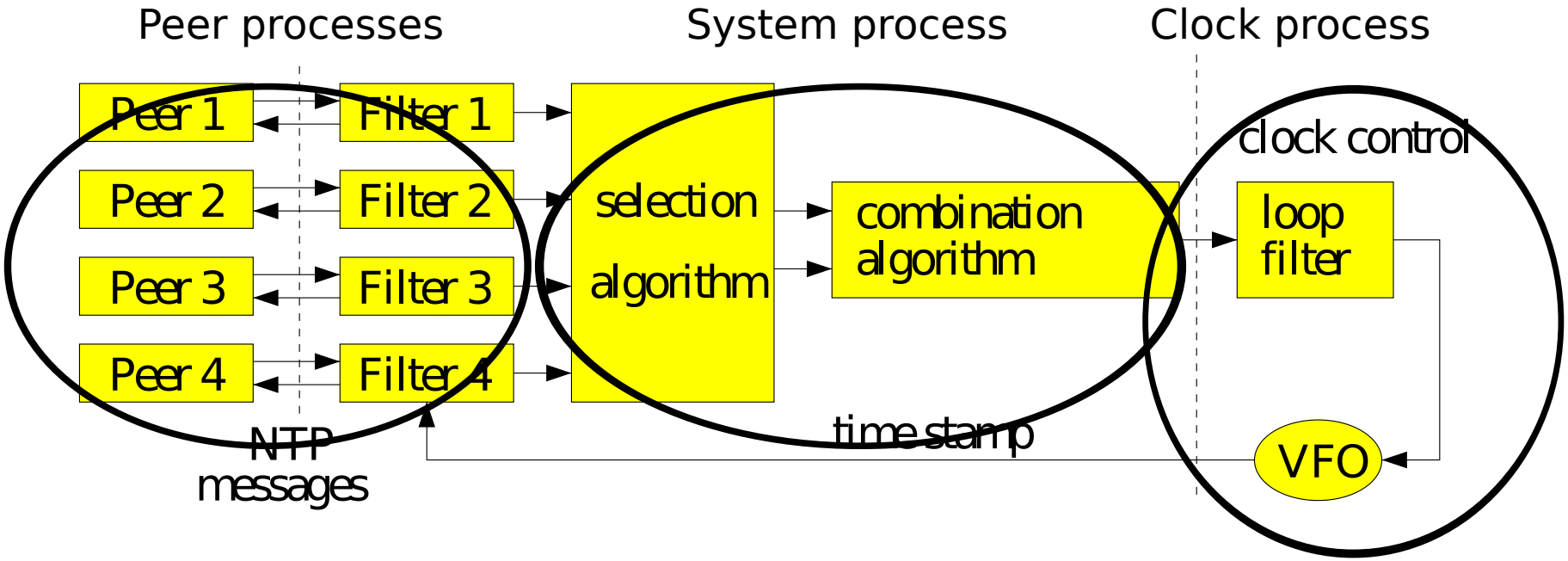


The Network Time Protocol – NTP

Architecture overview

- Multiple reference servers (“peers”) for redundancy and error diffusion
- Peer filters: select best value of last 8 offset readings for each reference server
- Filter into correct clocks (“truechimers”) and wrong clocks (“falsetickers”), then select reference clock that yields a high precision
- Combination algorithm: weighted average of the clock offsets (older NTP versions: value of reference node that seems to be most trustworthy)
- Loop filter and VFO: controlled local clock; minimize jitter and drift in the control loop

The Network Time Protocol – NTP processes

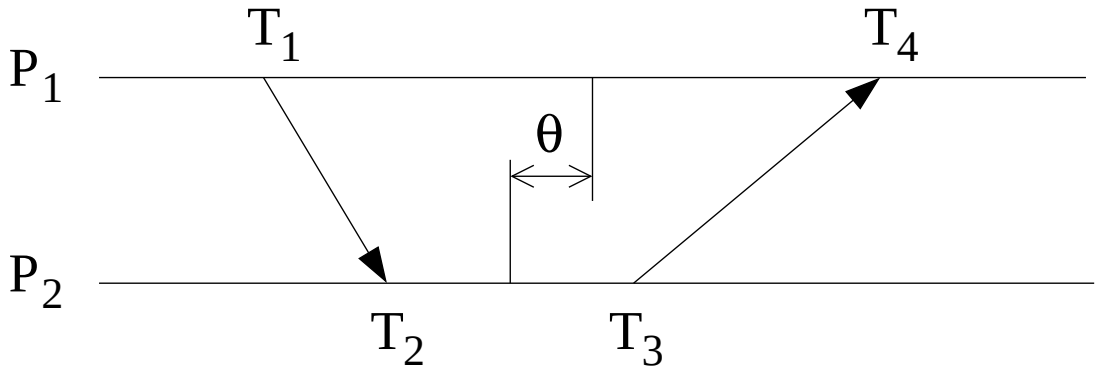


- Peer processes: independent and periodic; frequency determined by system process and remote servers
- System process: Periodic, frequency determined by phase jitter and stability of the local clock
- Clock process: Periodically in 1s-intervals (control VFO phase and frequency)

The Network Time Protocol – NTP

Offset measurement:

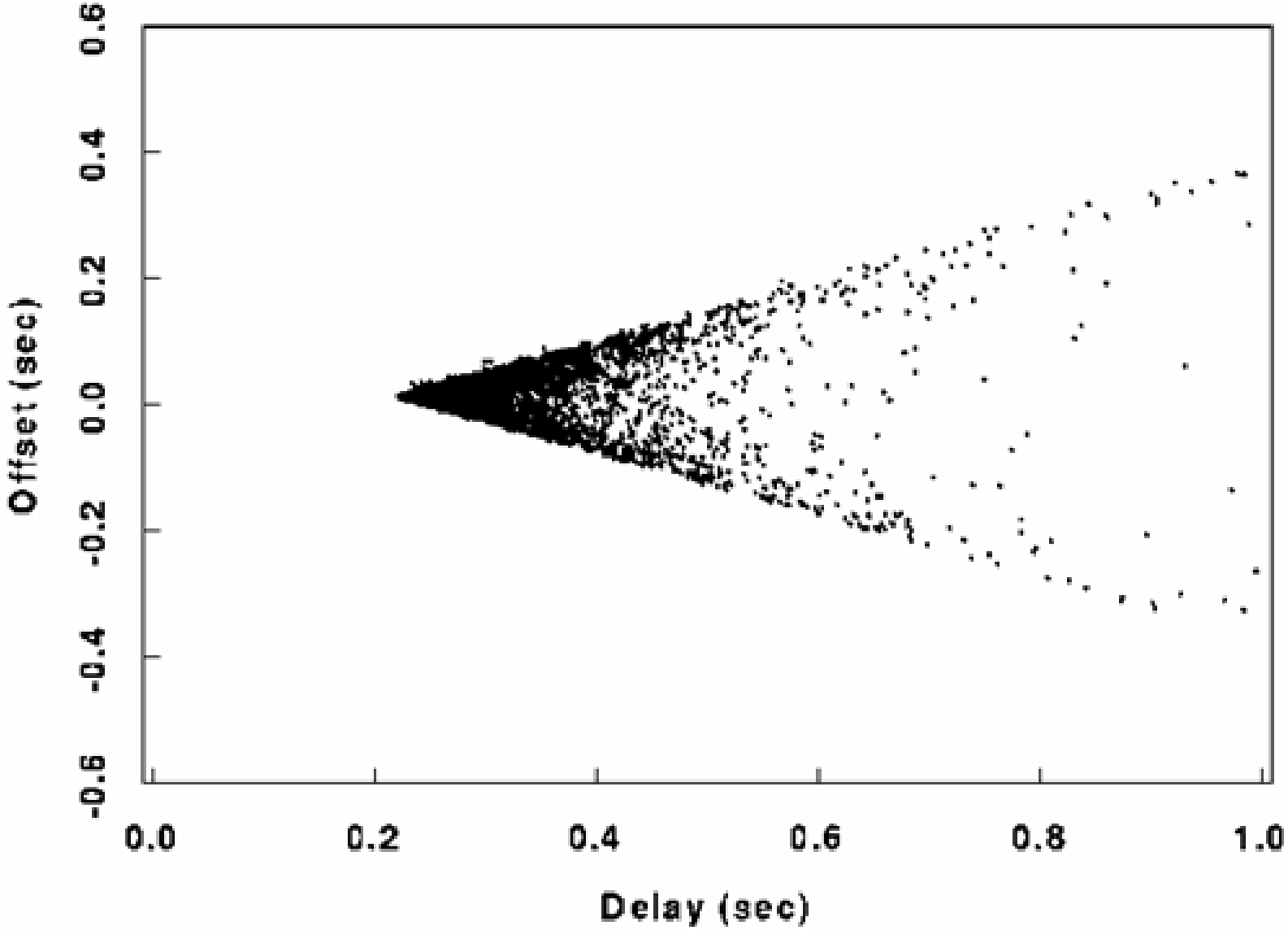
- Basic bidirectional method:



- Message transmission time: $\delta = (T_4 - T_1) - (T_3 - T_2)$
- Offset estimation: $\theta = (T_2 + T_3)/2 - (T_1 + T_4)/2$
- NTP message always includes $T_1 \ldots T_4$ (filled with values during round-trip communication)

- The last 8 readings are stored in buffer
  (following formulas: index $i = 0$: latest measurement)

The Network Time Protocol – NTP



The Network Time Protocol – NTP

Peer filtering algorithm: input data

- For each measurement, these values are obtained:
  - Offset θ, delay δ
  - Error estimation:
    $\epsilon =$ read accuracy + MAXDRIFT $*(T_4 - T_1)$
- Measurements are stored into a buffer
  - Storage for the last 8 values
  - Error estimate ϵ is updated after each measurement to reflect possible errors due to ageing (clock drift with time)

$i = 7 \ldots 1$: $(\delta_i, \theta_i, \epsilon_i) =$ $(\delta_{i-1}, \theta_{i-1}, \epsilon_{i-1} +$ MAXDRIFT $*$ elapsed time $)$

$i = 0$: $(\delta_0, \theta_0, \epsilon_0) =$ new reading

The Network Time Protocol – NTP

Peer filtering algorithm: filtering

- Input: list of measurement values $(\delta_i, \theta_i, \epsilon_i)$
- Sort by value of $\epsilon_j + \delta_j/2$
  Minimum value is used as a reference
- Calculate the filter scattering ϵ_ρ as weighted deviation of the measured values from the references (i.e. its a measure of the accuracy of the measurements)

  $$\epsilon_\rho = \sum_{i=0}^{7} |\theta_i - \theta_0| \, v^i \text{ with } v = 0.5 \text{ (experimentally choosen)}$$
- The result for each peer is a triple of delay, offset and dispersion
  $(\delta, \theta, \epsilon) = (\delta_0, \theta_0, \epsilon_0 + \epsilon_\rho)$
- NTPv4: Additionally, calculating the variance of the measured values

The Network Time Protocol – NTP

Calculation of variables for the system process:

- For each peer, the following variables are calculated:
  - $\Theta = \theta$
  - $\Delta =$ RootDelay $+ \delta$
  - $E =$ RootDispersion $+ \epsilon +$ (age of measurement) * MAXDRIFT
  - $\Rightarrow$ Triple $(\theta_i, \Delta_i, E_i)$ for each peer i
- System process receives these values and calculates
  - Synchronization distance $\Lambda_i = E_i + \Delta_i/2$
  - Correctness interval: $[\Theta_i - \Lambda_i, \Theta_i + \Lambda_i]$

The Network Time Protocol – NTP

Selection algorithm:

- Separation of “truechimers” and “false tickers”
- DTS (Digital Time Service, simple predecessor of NTP):
  - Compute interval in which a maximum number of correctness intervals overlap
  - Center of that interval is used as an offset for the clock correction
- Goal of NTP: Selection of an interval so that the center points of the intervals that are considered correct are within that interval

The Network Time Protocol – NTP

Clustering algorithm

- Goal: Further selection of the “best” references
  (stratum and synchronization distance as small as possible)
- Sorting the list of references according to the metric
  stratum * MAXDISPERSE + synchronization distance
- For each peer k, calculate weighted scatter value $\epsilon_{\xi,k}$

  $$\epsilon_{\xi,k} = \sum_{i=0}^{n} |\theta_i - \theta_k| \, w^i \text{ with } w = 0.75 \text{ (experimental)}$$
- The node with maximum $\epsilon_{\xi,k}$ is removed, and the whole of the process is repeated as long as
  - there are more than MINLOCK nodes on the list
  - the maximum $\epsilon_{xi,k}$ greater than the minimum E_i

The Network Time Protocol – NTP

Combination algorithm

- The remaining nodes are used for synchronization
  - Simple version: If the previously used reference nodes is on the list, keep using this references for synchronization, otherwise select the node in top of the list as a new synchronization references
  - Complexes variant (NTPv4): calculate a weighted average of the offsets of all nodes on the list
- The offset obtained this way is passed to the clock control (clock discipline algorithm)

The Network Time Protocol – NTP

Clock control: the clock discipline algorithm

- Control principle: Hybrid PLL / FLL control loop
- Aim of the hybrid scheme:
  - Frequency control for compensating frequency errors and long-term fluctuations
  - Phase control for correcting of short-term fluctuations

Summary

- Time stamps are important for many tasks in distributed systems
- Logical clocks
  - Can be used to order events, in particular in asynchronous systems
  - Two important examples: Lamport clocks, vector clocks
- Physical clocks
  - have limited accuracy due to technical reasons
  - synchronization can reduce the offset between local clocks
  - Basic task: offset measurement
  - Several synchronization algorithms exist; most important one: NTP