37th Meeting of the IFIP Working Group 10.4
Invited Talk at Workshop on *Time and Dependability*
January 21–25, 2000, Martinique (France)

High-Accuracy Time Services and Fault-Tolerant Clock Synchronization

Ulrich Schmid

Technische Universität Wien

Department of Automation
Treitlstraße 1, A-1040 Vienna
s@auto.tuwien.ac.at

January 27, 2000

La Carte

Motivation

Why is GPS + NTP ≠ all that we need?

♢

Fault-tolerant external clock synchronization

Traditional approaches

Interval-based approach

♢

How to achieve high accuracy?

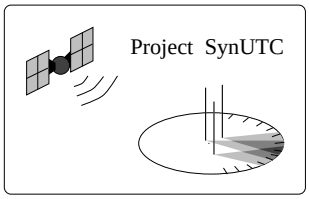
Rate synchronization

Timestamping

Clock design

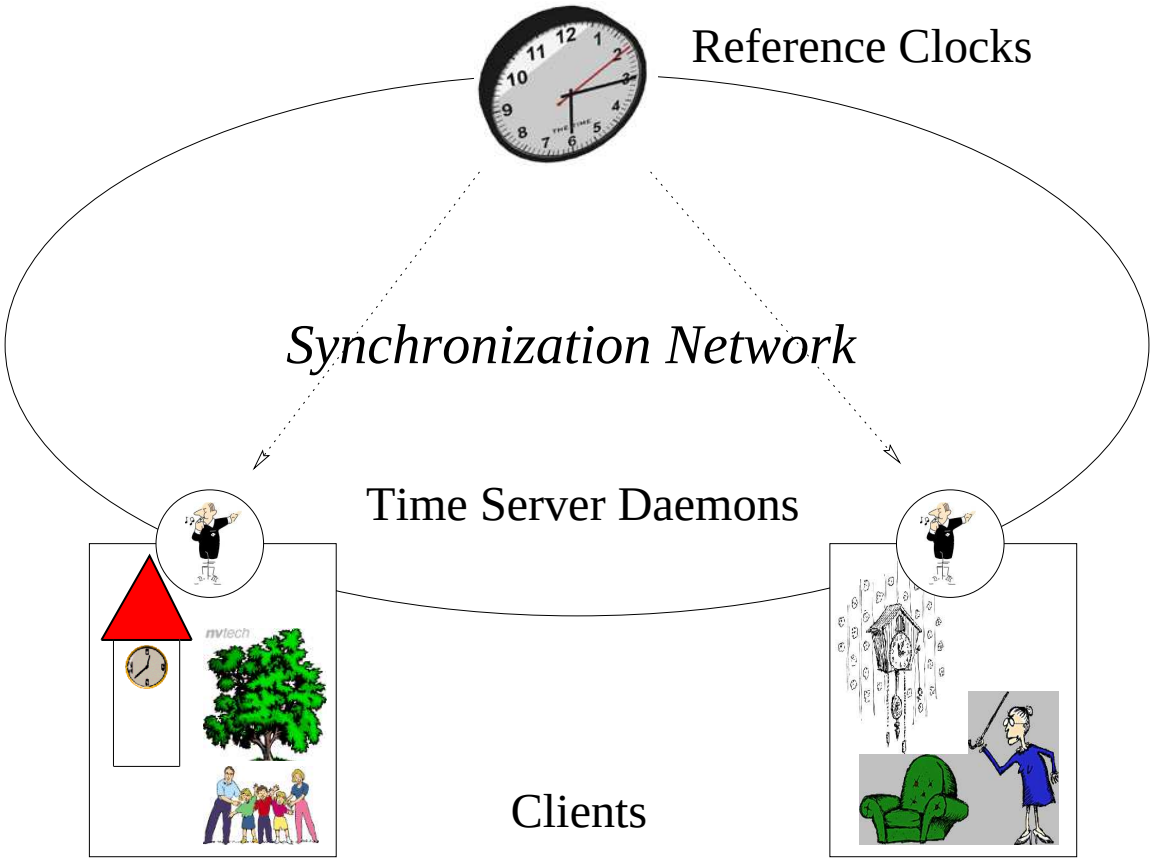
♢

References



http://www.auto.tuwien.ac.at/Projects/SynUTC

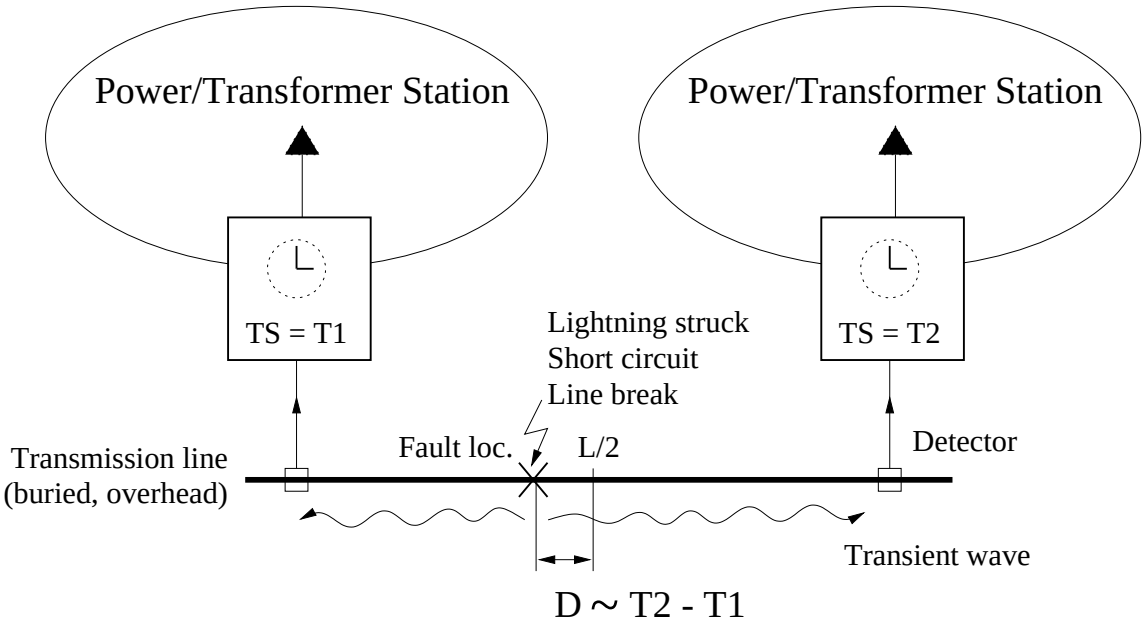
Time Services



Time service supplies common notion of time to local clients:

- Reference clocks know right time
- Server daemons consult reference clocks and adjust local clocks accordingly

Application: Fault Location in Power Distribution Grids



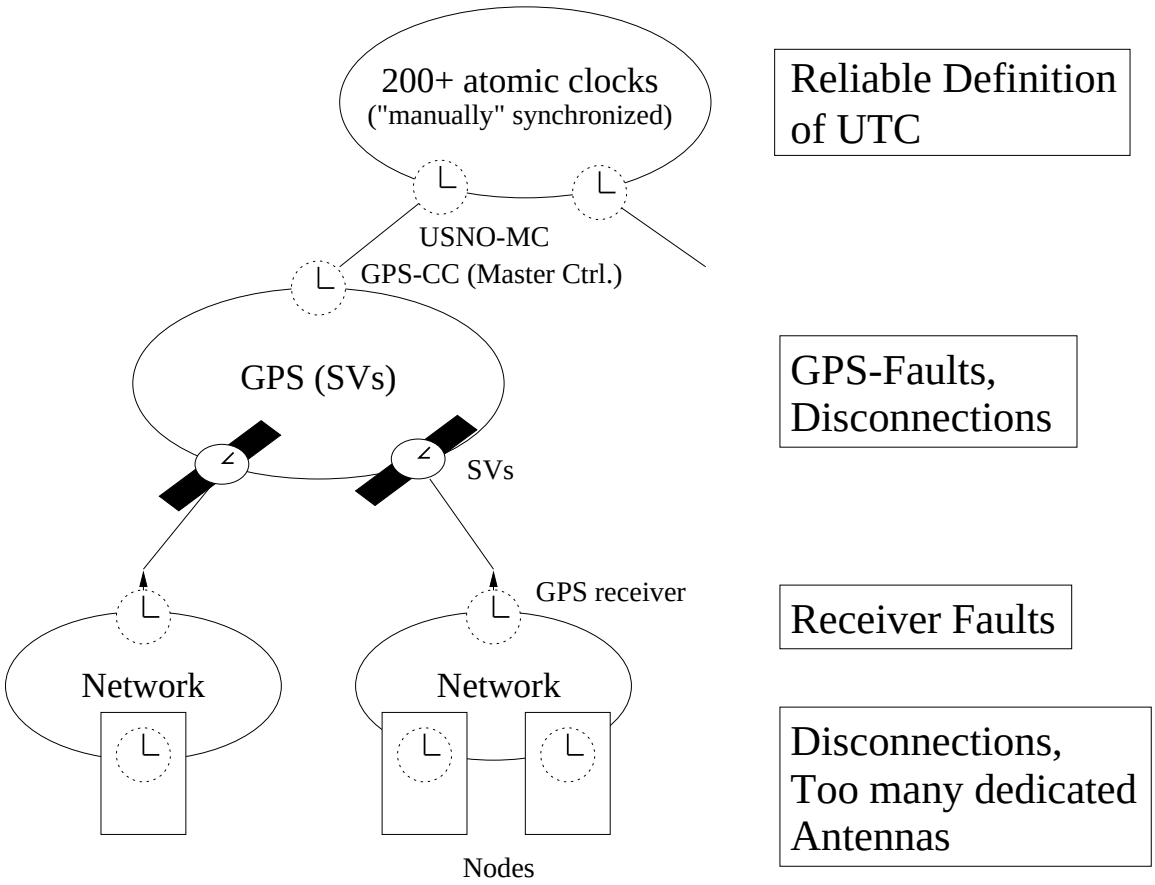
Demanding requirements:

- Fault location within ±10 m-range ⇒ 10 ns-range accuracy
- Up to 100+ endpoints in large power/transformer stations
- Cable length L up to 100 km
- Bad environmental conditions

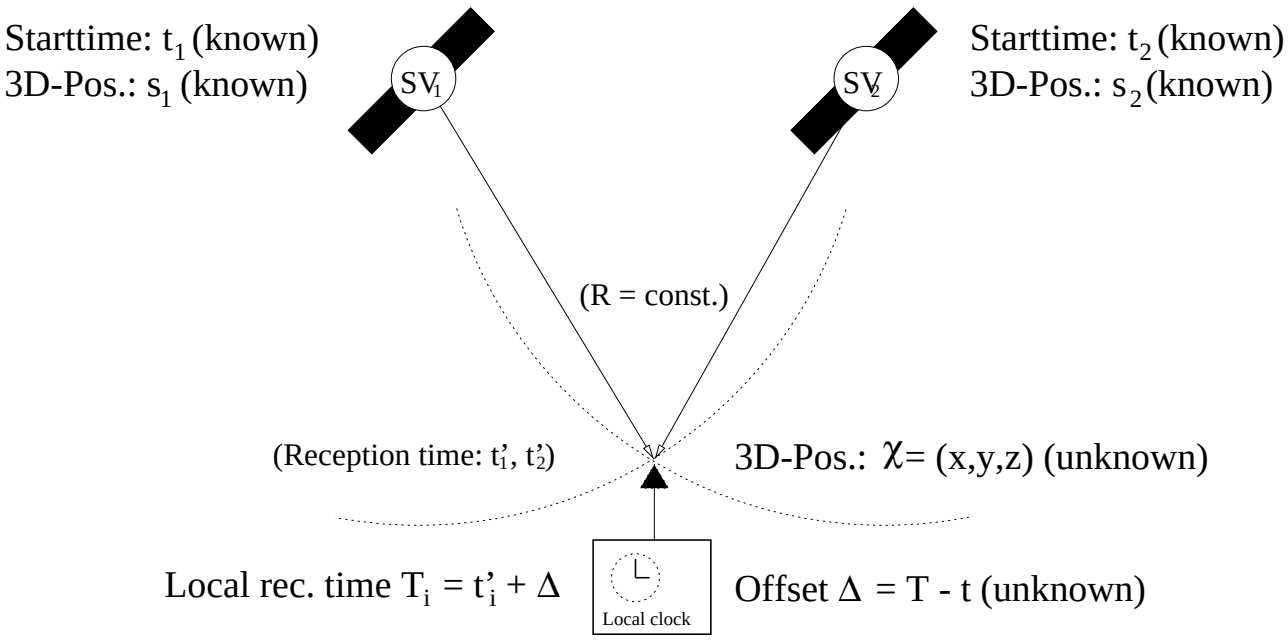
Why Clock Synchronization at all?

Often heard questions:

- Why not simply implement time service by dedicated GPS receivers at all nodes?
- Why bother with clock synchronization at all?



GPS (1)



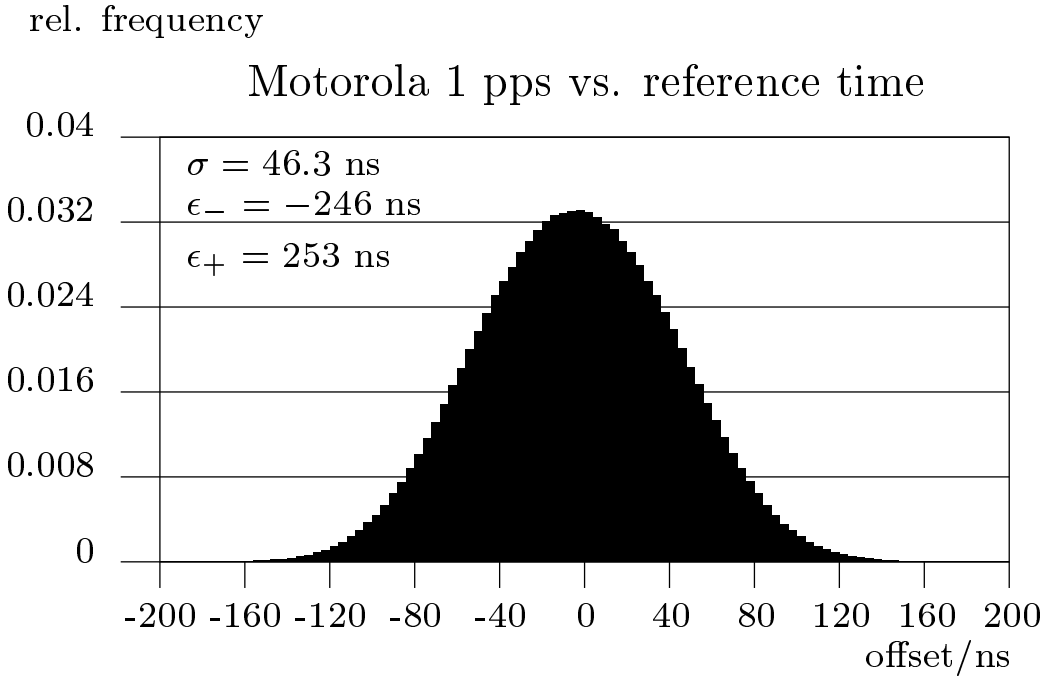
GPS-Receiver solves system of equations:

$$t_1 + |\chi - s_1|/c + \Delta = T_1$$
$$t_2 + |\chi - s_2|/c + \Delta = T_2$$
$$t_3 + |\chi - s_3|/c + \Delta = T_3$$
$$t_4 + |\chi - s_4|/c + \Delta = T_4$$

with

- 4 satellites required to determine (x, y, z) and Δ
- 1 satellite sufficient for Δ if (x, y, z) is known
- RAIM (Receiver Autonomous Integrity Monitoring)

GPS Features



Modular GPS timing receivers:

+ 100 ns-range accuracy (10 ns-range DGPS/CV)
+ Less than credit-card size
+ Cost < US$ 50,–
+ Digital 1 pps signal
+ RS-232 time+status info
+ Optional 10 MHz output
– Need dedicated rooftop antenna!
– Time-to-fix up to 30 minutes!

Reliability GPS

Experimental evaluation of 6 GPS receivers [HS97] revealed

- 1 pps omissions
- spurious 1 pps and step errors (= delivered time erroneously jumps)
- ramp errors (= delivered time erroneously drifts away)
- "Y2K problems"

Other problems [Dan97], [HS97]

- Receiver firmware errors
- Complex GPS errors modes
- Current GPS has single points of failure
- Dependency upon a single system questionable

"Pure" GPS-based Time Service

Score:

+ Excellent accuracy (100 ns and below)
+ Usually very reliable operation ([HS97]: 10^{-6})
– Every node's GPS receiver needs rooftop antenna
– Exhaustive assessment of all error modes impossible
– Possibly large delay until correct time is provided

Hence,

- large number of nodes creates "forest" of antennas
- GPS alone not sufficient for provably correct, dependable systems
- fast joining of nodes cannot be guaranteed

⇒ **GPS is great, but no panacea ...**

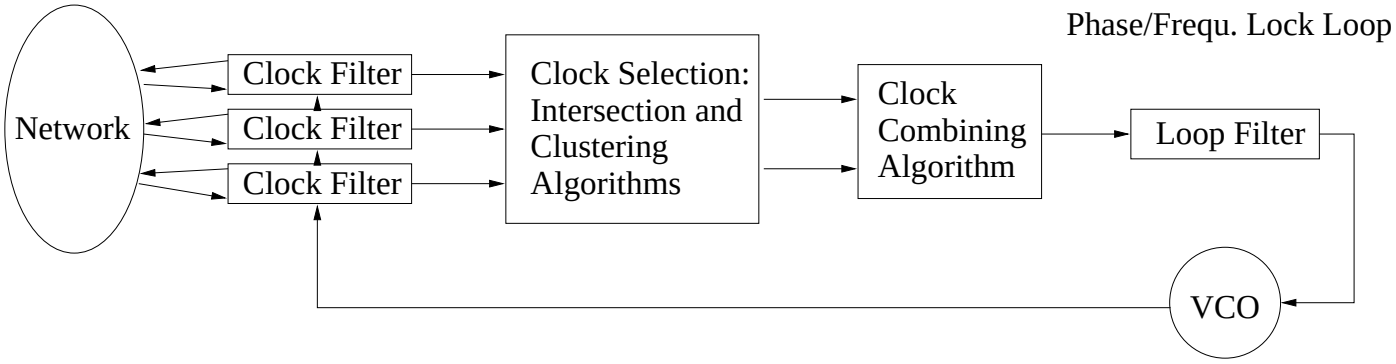
NTP (1)

So why not using a combination of GPS and NTP?

NTP [Mil91] employs

- distributedly maintened minimum-weight spanning tree of time servers (= reference clocks)
- remote clock reading of all peer time servers
- well-engineered statistical algorithms for data filtering and clock selection.

Overall structure ([Mil95]):



NTP (2)

Score of NTP in "nice" system architectures:

+ Readily available, standardized and field-proven
  ⇒ **industry jumps at it!**

– Complex algorithms designed for "bad" (asynchronous) settings like the Internet ⇒ complex error modes

– Behavior in "nice" (synchronous) systems not investigated

– Designed for ms-range accuracy only

– Delivery of (non-chronoscopic) UTC

Hence,

- NTP cannot forward GPS accuracy to clients
- deterministic analysis not available
- NTP not suitable for provably correct, dependable systems

⇒ **Blindly jumping at NTP is not advisable . . .**

Fault-Tolerant (External) Clock Synchronization

We consider networked distributed systems with

- $r \geq 0$ nodes hosting reference clocks $R_i(t)$
- $n > 0$ client nodes with local clocks $C_i(t)$
- maximum clock drift ρ
- maximum transmission delay uncertainty ε
- suitable fault model

Clock synchronization algorithm must guarantee

- internal synchronization with precision π

$$|C_i(t) - C_j(t)| \leq \pi$$

- external synchronization with accuracy α

$$\text{Sufficiently many } R_i(t) \text{ correct} \Rightarrow |C_i(t) - t| \leq \alpha$$

Clock Synchronization Algorithms

Many different algorithms exist, but can be explained by a generic principle [Sch86]:

1. Generate (approximately) simultaneous event at all nodes

- exploit synch. clocks: [LWL88], [FC97b], SynUTC
- generate event via messages: [ST87], [VRC97]

2. Read remote clocks' values

- one-way: [LWL88], SynUTC
- round-trip: [Cri89]

3. Compute convergence function

- Fault-Tolerant Midpoint [LWL88], [Sch97b]
- Optimal Precision Algorithm [FC95], [Sch97a]

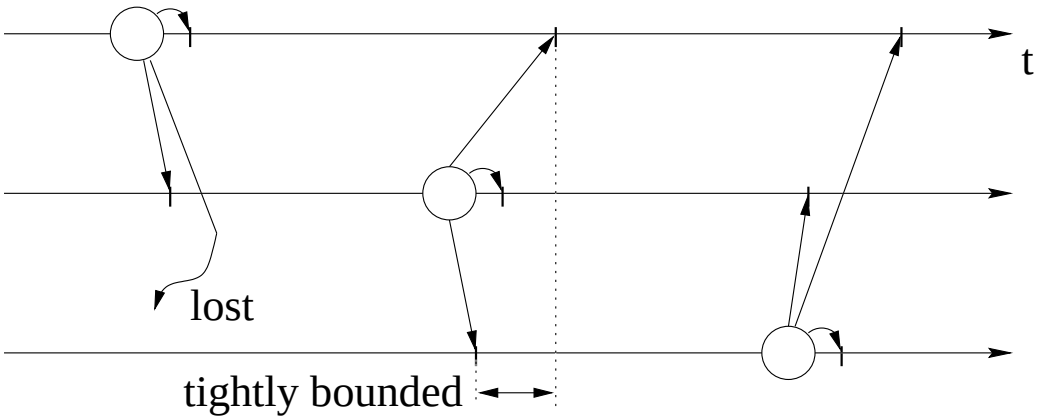
4. Adjust local clock

- instantaneously
- continuous amortization [SC90], [SS97a]

Example: CesiumSpray / A Posteriori Agreement

Assume

- a network with HW-broadcasting capability
- at least one of $f + 1$ broadcasts heared by all nodes



A posteriori agreement: Any node

1. periodically broadcasts start message
2. creates a virtual clock (vc) upon arrival of any start message and broadcasts local clock reading

3a. detects+selects candidate faultless broadcast

3b. computes accuracy-preserving clock adjustment from clock readings (assuming own candidate will win)

3c. runs agreement protocol to select winner

4. applies adjustment and switches to winning vc

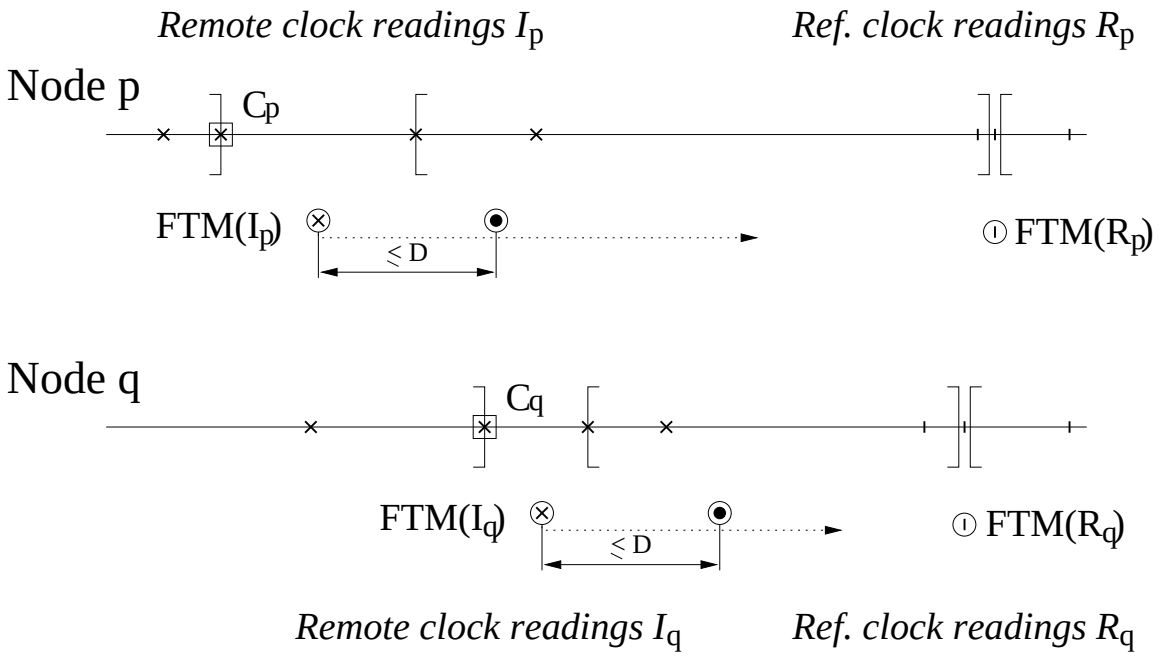
Example: FTM/Optimal + External Algorithm

Any node

1/2. reads remote clocks when local time $= kP$, $k \geq 1$

3. applies convergence function to clock readings

4. adjusts local clock



- [FC97b]: Integration of external time by shifting at most $D \approx P\rho$ towards reference clock readings
- [FC95]: Optimal precision $\pi = 4\varepsilon + 4P\rho$ by enforcing nested adjustment condition

Faults of Reference Clocks

External/internal algorithm [FC97b]

- Always guarantees internal synchronization condition
- Maintains external synchronization condition if at most $r/2$ reference clocks exhibit arbitrary faults
- Primary problem: slow ramp errors below ρ

$\Rightarrow$ Alg is optimal according to lower-bound theorem [FC97b]

$\Rightarrow$ No algorithm can do better in this respect

Still, there is room for improvement:

- Add accuracy-awareness ("fail-awareness" [FC97a])
- Handle "more evident" reference clock faults explicitly
  - $\Rightarrow$ ensure some QoS for omissions & step errors
  - $\Rightarrow$ decreasing ρ increases fault coverage w.r.t. ramp errors

$\Rightarrow$ **SynUTC research project**
http://www.auto.tuwien.ac.at/Projects/SynUTC

Interval-Based Clock Synchronization

Basic idea [Mar84]: Replace ordinary clocks $T = C(t)$ by interval clocks

$$\boldsymbol{C}(t) = [C(t) - \alpha^-(t), C(t) + \alpha^+(t)]$$

incorporating

- a usual local clock $C(t)$
  $\Rightarrow$ local time information
- an interval of accuracies $\boldsymbol{\alpha}(t) = [-\alpha^-(t), \alpha^+(t)]$
  $\Rightarrow$ dynamically maintained accuracy bounds

managed appropriately to satisfy the

- internal synchronization condition

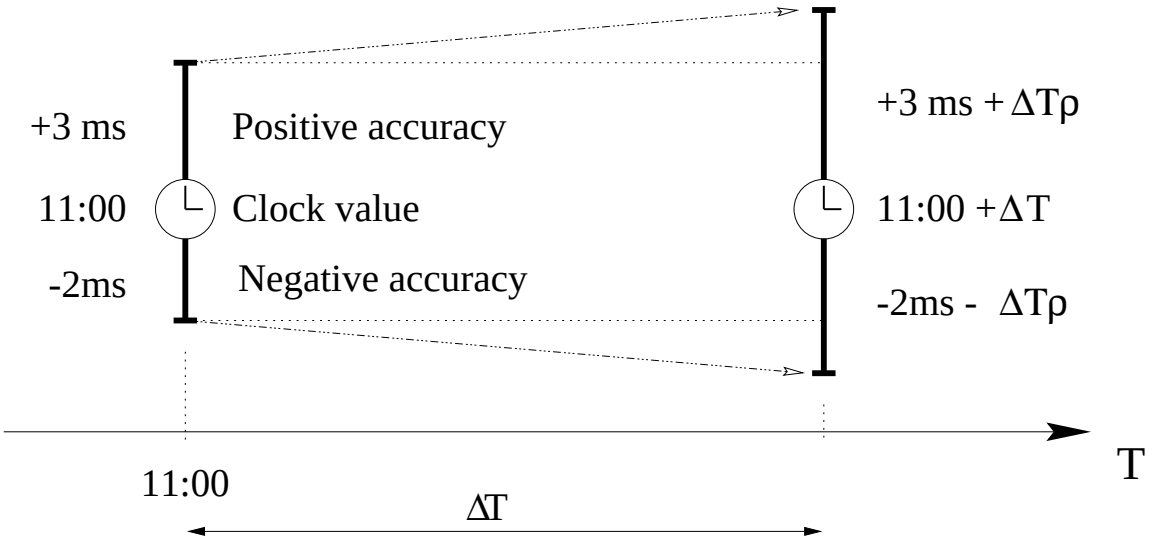
$$|C_p(t) - C_q(t)| \leq \pi \qquad \text{for some fixed } \pi$$

- generalized external synchronization condition

$$t \in \boldsymbol{C}(t) \quad \Longleftrightarrow \quad -\alpha_p^+(t) \leq C_p(t) - t \leq \alpha_p^-(t)$$

Basic Building Blocks

Accuracy-preserving enlargement of intervals:



Drift compensation:

- ΔT defined by some desired local time
- Deterioration proportional to ΔT
- Incorporating granularity effects easy

Delay compensation:

- ΔT defined by transmission delay
- Uncertainty ε instead of deterioration

Generic Algorithm [SS97a]

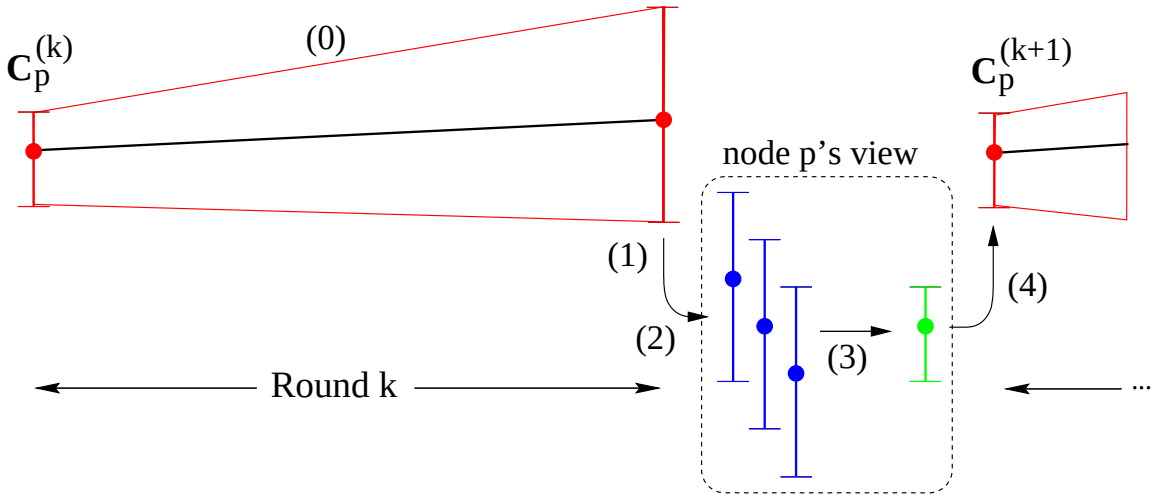
Any node

(0) deteriorates its local interval clock

(1/2) reads remote interval clocks when local time $= kP$, $k \geq 1$

(3) applies interval-based convergence function

(4) adjusts local interval clock



Convergence functions:

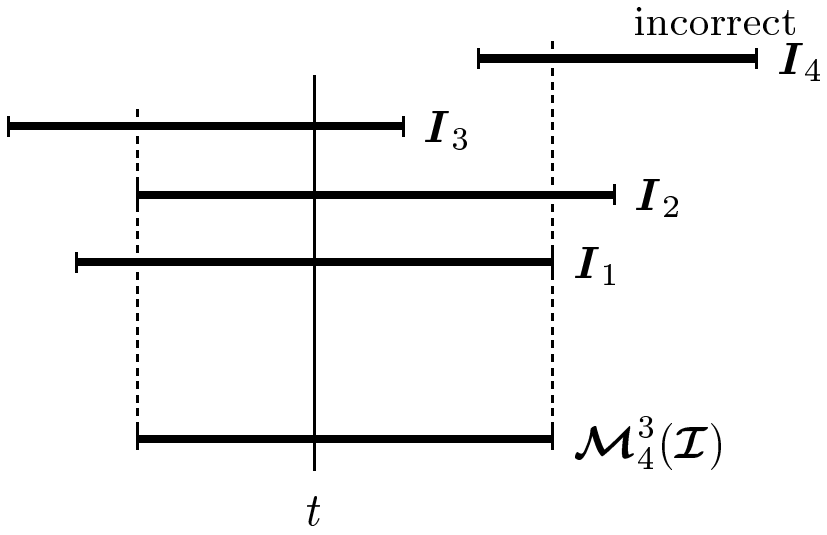
$\mathcal{OA}$ [Sch97b]	$\simeq$	FTM [LWL88]
$\mathcal{OP}$ [Sch97a]	$\simeq$	Optimal alg. [FC95]
Clock validation [Sch95]	$\sim$	Ext/int alg. [FC97b]

Fault-tolerant Interval Intersection

Given n input intervals

- all supposed to contain t
- some of them being faulty

how to find a minimal interval containing t?



- Marzullo-function $\mathcal{M}$ [Mar84]: The largest interval whose endpoints intersect at least $n-f$ input intervals
- Fault-Tolerant Interval Function $\mathcal{F}$ [SS99]: The interval formed by the $f+1$-largest left edge resp. the $f+1$-smallest right edge of the n input intervals

$\mathcal{F}$ satisfies Lipschitz condition $\Rightarrow$ settles open problem from [Lam87]

Generic Precision & Accuracy Analysis

Generic precision analysis [SS97a],[SS97b]

- Accuracy and precision treated independently
- Precision analysis reduced to "accuracy analysis" w.r.t. artificial internal global time τ

$\Rightarrow$ utilize the same tools for accuracy & precision analysis:

	Accuracy analysis	**Precision analysis**
	real-time t	internal global time $\tau = \tau(t)$
	accuracy intervals $\boldsymbol{\alpha}(t)$	precision intervals $\boldsymbol{\pi}(\tau)$
	$t \in C(t) + \boldsymbol{\alpha}(t)$	$\tau \in C(t) + \boldsymbol{\pi}(\tau)$
$\mathcal{OA}$:	$\mathcal{M}$, $\mathcal{F}$	$\mathcal{M}$, $\mathcal{F}$, FTM
$\mathcal{OP}$:	$\mathcal{M} \cap \boldsymbol{C}$, $\mathcal{F} \cap \boldsymbol{C}$	$\mathcal{M} \cap \boldsymbol{C}$, $\mathcal{F} \cap \boldsymbol{C}$, Opt.alg

Analysis of $\mathcal{OA}$ [Sch97b] and $\mathcal{OP}$ [Sch97a]

- Hybrid fault model (crash/benign/arbitrary)
- Detailed system model (granularities, latencies)
- Surprisingly generic results

$\Rightarrow$ Very accurate, widely applicable formulas

Clock Rate Synchronization

Interval-based algs need parameters like ρ explicitly:

1. A priori values
2. Online measurement (+ control)

Clock state synchronization	**Clock rate synchronization**
$C_p(t)$	$v_p(t) = \frac{d\,C_p(t)}{d\,t}$
ideal: $C_p(t) = t$	ideal: $v_p(t) = 1$
Clock drift ρ	Clock stability σ
Precision π:	Consonance γ:
$\lvert C_p(t) - C_q(t)\rvert \leq \pi$	$\lvert v_p(t) - v_q(t)\rvert \leq \gamma$
Accuracy α:	Drift ρ:
$\lvert C_p(t) - t\rvert \leq \alpha$	$\lvert v_p(t) - 1\rvert \leq \rho$

Implementation:

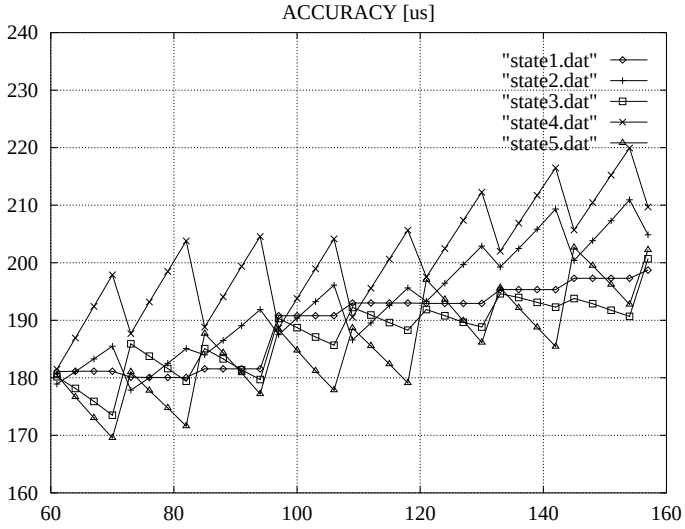
- "Remote clock rate reading" can easily be piggybacked on clock state synchronization traffic
- Apply interval-based fault-tolerant rate synchronization algorithms

$\Rightarrow$ Enables high-precision clock synchronization with low-precision clocks

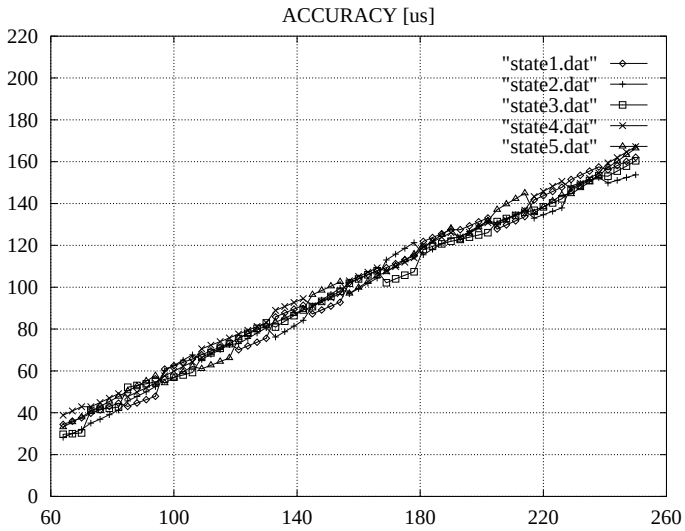
Results Clock State+Rate Synchronization

Simulation results for 5 nodes (SimUTC-toolkit [WGSS99])

Accuracy $\alpha_p(t) = C_p(t) - t$ for FTM state only:



Accuracy $\alpha_p(t) = C_p(t) - t$ for FTM state + rate:



High-Accuracy Clock Reading

Two basic clock reading methods:

(1) One-way time transfer

+ delivers minimimal uncertainty
– needs δ, ε securing transmission delays $\delta' \in [\delta \pm \varepsilon]$
– assumption coverage in real systems?

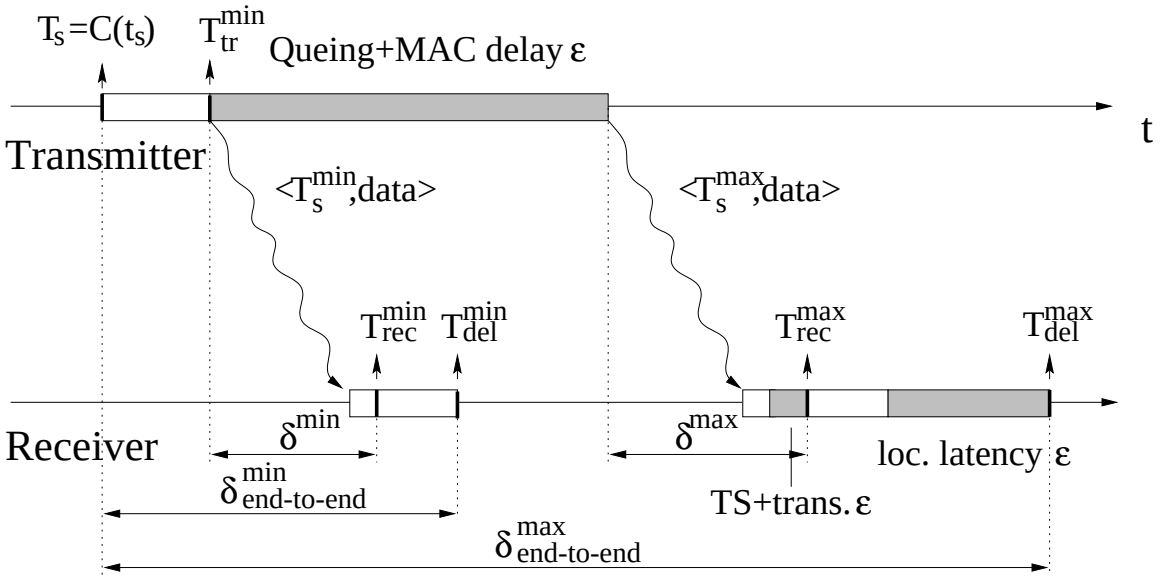
(2) Round-trip clock reading method [Cri89]

+ provides remote clock values + error bounds
+ works in non-synchronous systems as well
– doubles uncertainty

Method (1) preferable for high accuracy clock synchronization

1. Transmitter p sends message $< T_s, data >$ at local time T_s
2. Receiver q gets it at local time T_{del} and concludes $C_q(t) - C_p(t) \in T_{del} - T_s - [\delta \pm \varepsilon]$

One-way Time Transfer in LANs



Drawing timestamps:

	Software-based	*Hardware-based*
Send TS	T_s	T_{tr}
Rec. TS	T_{del}	T_{rec}
Relev. δ	end-to-end	physical trans.
Clock read ε	$\delta^{\max}_{e-t-e} - \delta^{\min}_{e-t-e}$	$\delta^{\max} - \delta^{\min}$

Hardware Timestamping Support

Score:

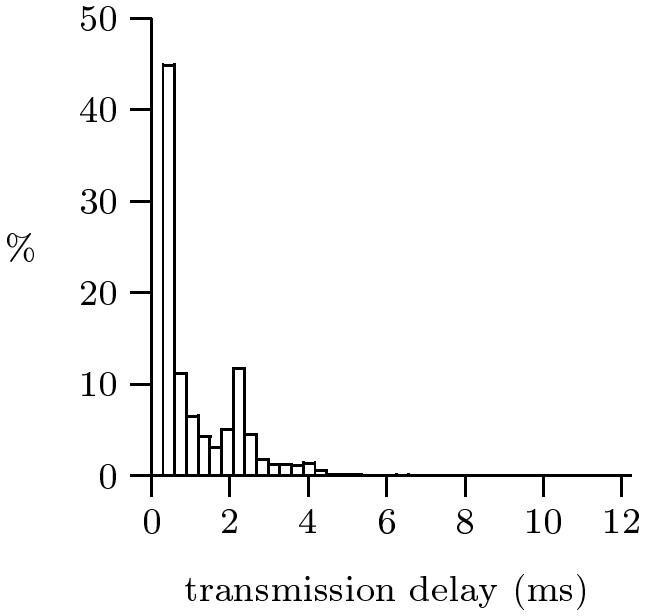
+ High-accuracy clock synchronization possible even in presence of very large (unbounded) end-to-end delays
+ Extends to WANs-of-LANs if all routers do HW-timestamping as well
+ Applicable in conjunction with round-trip clock reading $\Rightarrow$ On-line measurement of δ possible
+ Queueing&MAC delays and overloads do not affect ε
+ Needs only physical transmission delays
  $\Rightarrow$ Assessment and coverage analysis relatively easy
– Needs some special-purpose hardware

Available implementations:

- Memory-based timestamping [KO87], [SKM+00]
  Network Time Interface (NTI): $\varepsilon \approx \mu$s-range
- MII-based timestamping for (Fast) Ethernet [SHK99]:
  $\varepsilon \approx 10$ ns-range
- Network-controllers exporting trigger signals:
  [KKMS95] (TTP), [HL98] (LON)

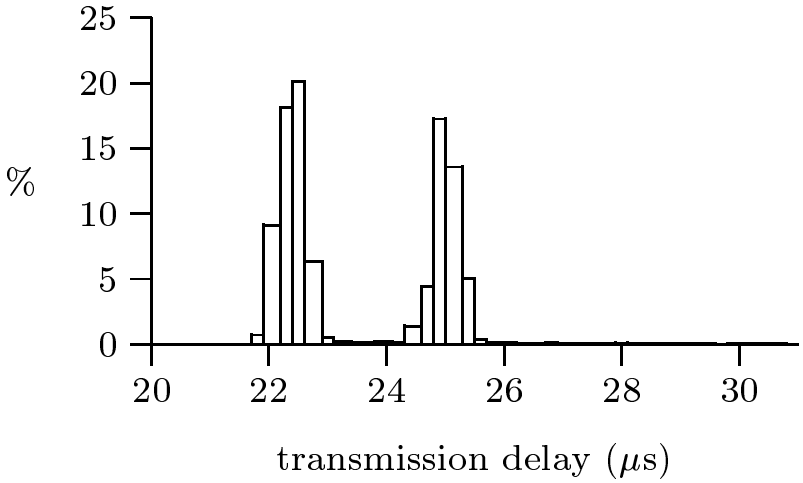
Measurement results [SKM+00]

Histogram of end-to-end delays (Ethernet):



Minimum	0.30 ms
99%-minimum	0.30 ms
95%-minimum	0.30 ms
95%-maximum	4.17 ms
99%-maximum	9.24 ms
Maximum	37.00 ms
Average	1.35 ms
Std.Dev.	1.63 ms

Histogram of transmission delays (Ethernet + NTI):



Minimum	20.4 μs
99%-minimum	21.7 μs
95%-minimum	21.9 μs
95%-maximum	25.5 μs
99%-maximum	31.0 μs
Maximum	36.8 μs
Average	23.6 μs
Std.Dev.	1.6 μs

High-Accuracy Clock Design

Rigorous treatment of granularities [SS97a] identified

- clock (reading) granularity G
- clock setting granularity $G_S \leq G$
- discrete rate adjustment uncertainty u (usually $u = 1/f_{osc}$)

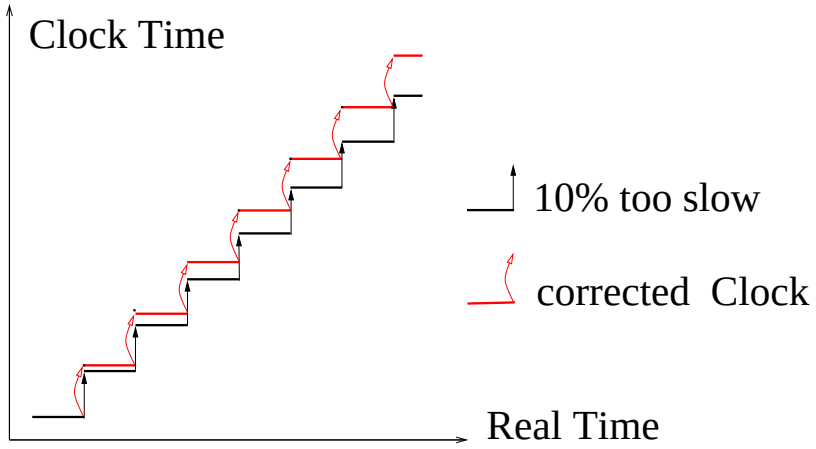
Detailed worst-case analysis [Sch97b], [Sch97a] showed

$$\begin{aligned} \pi_{FTM} &\approx 5\varepsilon + 4P\rho + 4G + 12u + G_S \\ \pi_{Opt} &\approx 4\varepsilon + 4P\rho + 3G + 11u + G_S \end{aligned}$$

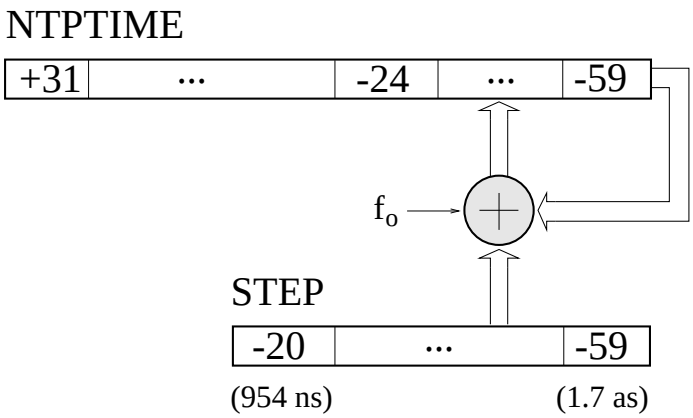
Hence, high-accuracy clock synchronization requires

- high-resolution clock
- high-frequency oscillator
- high-resolution clock adjustment capabilities

Example: Adder-Based Clock [SSHL97]



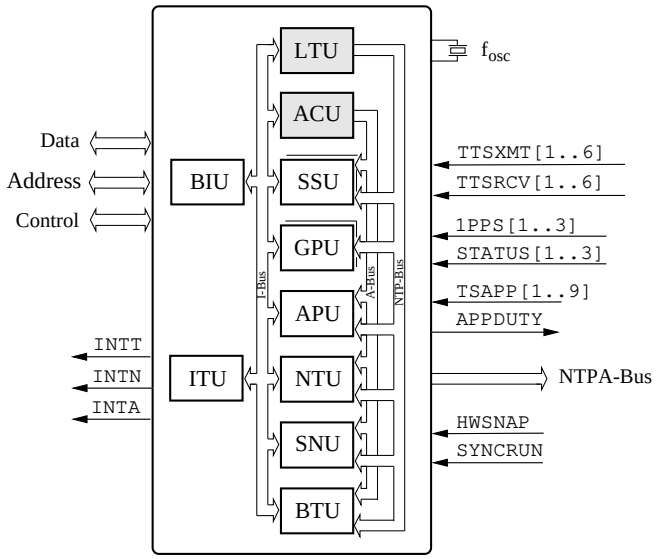
Adder-based clock design



Features:

- Can use arbitrary oscillator frequency 1 . . . 25 MHz
- fine-grained clock rate adjustment (10 ns/s)
- continuous amortization and leap second insertion/deletion in hardware

UTCSU-ASIC



unit	full name
ACU	Accuracy Unit
APU	Application Unit
BIU	Bus Interface Unit
BTU	Built-In Test Unit
GPU	GPS Unit
ITU	Interrupt Unit
LTU	Local Time Unit
NTU	Network Time Interface Unit
SNU	Snapshot Unit
SSU	Synchronization Subnet Unit

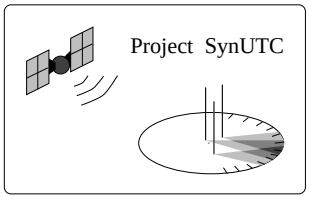
Features:

- Flexible 8/16/32-bit bus interface
- Adder-based local clock (LTU): 56-bit NTP-time + 8-bit checksum
- Local accuracy intervals (ACU) with automatic deterioration
- Atomic time+accuracy-stamping of 20+ digital input signals (GPS 1 pps etc.)
- Several Duty-timers
- On-line test and debugging features

Summary of SynUTC Accomplishments

Score interval-based clock state+rate synchronization:

+ Enables clock validation dealing with (most) reference clock faults
+ Enables accuracy-aware applications
  $\Rightarrow$ fail-awareness for (multi-cluster) applications
+ Incorporates a very detailed system model
+ Tight integration with clock rate synchronization
  $\Rightarrow$ low-precision local clocks sufficient
+ High-accuracy hardware support available
– Algorithms more complex
– Explicit bounds on some system parameters (ε, σ/ρ) required



http://www.auto.tuwien.ac.at/Projects/SynUTC

References

[Cri89] F. Cristian. Probabilistic clock synchronization. *Distributed Computing*, 3(3):146–158, 1989.

[Dan97] Peter H. Dana. Global Positioning System (GPS) time dissemination for real-time applications. *Real-Time Systems*, 12(1):9–40, January 1997.

[FC95] Christof Fetzer and Flaviu Cristian. An optimal internal clock synchronization algorithm. In *Proceedings 10th Annual IEEE Conference on Computer Assurance*, Gaithersburg, MD, June 1995.

[FC97a] C. Fetzer and F. Cristian. Fail-awareness: An approach to construct fail-safe applications. In *Proceedings of the 27th Annual International Symposium on Fault-Tolerant Computing*, Seattle, Jun 1997. http://www-cse.ucsd.edu/users/cfetzer/FAPS/faps.html.

[FC97b] Christof Fetzer and Flaviu Cristian. Integrating external and internal clock synchroniza-

tion. *J. Real-Time Systems*, 12(2):123–172, March 1997.

[HL98] Martin Horauer and Dietmar Loy. Hardware-unterstützte Uhrensynchronisation in Verteilten Systemen. In *Proceedings AUSTROCHIP'98*, pages 67–72, Wiener Neustadt, Austria, October 1998. (ISBN 3-901578-03-X, in German).

[HS97] Dieter Höchtl and Ulrich Schmid. Long-term evaluation of GPS timing receiver failures. In *Proceedings of the 29th IEEE Precise Time and Time Interval Systems and Application Meeting (PTTI'97)*, pages 165–180, Long Beach, California, December 1997.

[KKMS95] Hermann Kopetz, Andreas Krüger, Dietmar Millinger, and Anton Schedl. A synchronization strategy for a time-triggered multicluster real-time system. In *Proceedings Reliable Distributed Systems (RDS'95)*, Bad Neuenahr, Germany, September 1995.

[KO87] Hermann Kopetz and Wilhelm Ochsenreiter. Clock synchronization in distributed real-time systems. *IEEE Transactions on Computers*, C-36(8):933–939, 1987.

[Lam87] Leslie Lamport. Synchronizing time servers. Technical Report 18, Digital System Research Center, 1987.

[LWL88] Jennifer Lundelius-Welch and Nancy A. Lynch. A new fault-tolerant algorithm for clock synchronization. *Information and Computation*, 77(1):1–36, 1988.

[Mar84] Keith A. Marzullo. *Maintaining the Time in a Distributed System: An Example of a Loosely-Coupled Distributed Service*. PhD dissertation, Stanford University, Department of Electrical Engineering, February 1984.

[Mil91] David L. Mills. Internet time synchronization: The network time protocol. *IEEE Transactions on Communications*, 39(10):1482–1493, October 1991.

[Mil95] David L. Mills. Improved algorithms for synchronizing computer network clocks. *IEEE Transactions on Networks*, pages 245–254, June 1995.

[SC90] F. Schmuck and F. Christian. Continuous amortization need not affect the precision of a clock synchronization algorithm. In *Proceedings of the 9th Annual ACM Symposium on*

Principles on Distributed Computing (PODC), pages 133–144, Quebec City, Canada, August 1990.

[Sch86] Fred B. Schneider. A paradigm for reliable clock synchronization. In *Proceedings Advanced Seminar of Local Area Networks*, pages 85–104, Bandol, France, April 1986.

[Sch95] Ulrich Schmid. Synchronized Universal Time Coordinated for distributed real-time systems. *Control Engineering Practice*, 3(6):877–884, 1995. (Reprint from Proceedings 19th IFAC/IFIP Workshop on Real-Time Programming (WRTP'94), Lake Reichenau/Germany, 1994, p. 101–107.).

[Sch97a] Ulrich Schmid. Interval-based clock synchronization with optimal precision. Technical Report 183/1-78, Department of Automation, Technische Universität Wien, July 1997. (submitted to Information and Computation).

[Sch97b] Ulrich Schmid. Orthogonal accuracy clock synchronization. Technical Report 183/1-77, Department of Automation, Technische Universität Wien, March 1997. (submitted to

Chicago Journal of Theoretical Computer Science).

[Sch97c] Klaus Schossmaier. An interval-based framework for clock rate synchronization algorithms. In *Proceedings 16th ACM Symposium on Principles of Distributed Computing*, pages 169–178, St. Barbara, USA, August 1997.

[SHK99] Ulrich Schmid, Martin Horauer, and Nikolaus Kerö. How to distribute GPS-time over COTS-based LANs. In *Proceedings of the 31th IEEE Precise Time and Time Interval Systems and Application Meeting (PTTI'99)*, Dana Point, California, December 1999. (to appear).

[SKM+00] Ulrich Schmid, Johann Klasek, Thomas Mandl, Herbert Nachtnebel, Gerhard R. Cadek, and Nikolaus Kerö. A Network Time Interface M-Module for distributing GPS-time over LANs. *J. Real-Time Systems*, 18(1), 2000. (to appear).

[SS97a] Ulrich Schmid and Klaus Schossmaier. Interval-based clock synchronization. *J. Real-Time Systems*, 12(2):173–228, March 1997.

[SS97b] Ulrich
Schmid and Klaus Schossmaier. Interval-based clock synchronization revisited. Technical Report 183/1-80, Technische Universität Wien, Department of Automation, July 1997.

[SS99] Ulrich Schmid and Klaus Schossmaier. How to reconcile fault-tolerant interval intersection with the Lipschitz condition. Technical Report 183/1-96, Department of Automation, TU Vienna, September 1999. (submitted to Distributed Computing).

[SSHL97] Klaus Schossmaier, Ulrich Schmid, Martin Horauer, and Dietmar Loy. Specification and implementation of the Universal Time Coordinated Synchronization Unit (UTCSU). *J. Real-Time Systems*, 12(3):295–327, May 1997.

[ST87] T. K. Srikanth and Sam Toueg. Optimal clock synchronization. *Journal of the ACM*, 34(3):626–645, July 1987.

[VRC97] Paulo Veríssimo, Luís Rodrigues, and Antonio Casimiro. CesiumSpray: a precise and accurate global clock service for large-scale systems. *J. Real-Time Systems*, 12(3):243–294, 1997.

[WGSS99] Bettina Weiss, Günther Gridling, Ulrich Schmid, and Klaus Schossmaier. The SimUTC fault-tolerant distributed systems simulation toolkit. In *Proceedings 7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'99)*, pages 68–75, Univ. of Maryland, USA, October 1999.