Globale Zeit und Uhrensynchronisation in Netzen

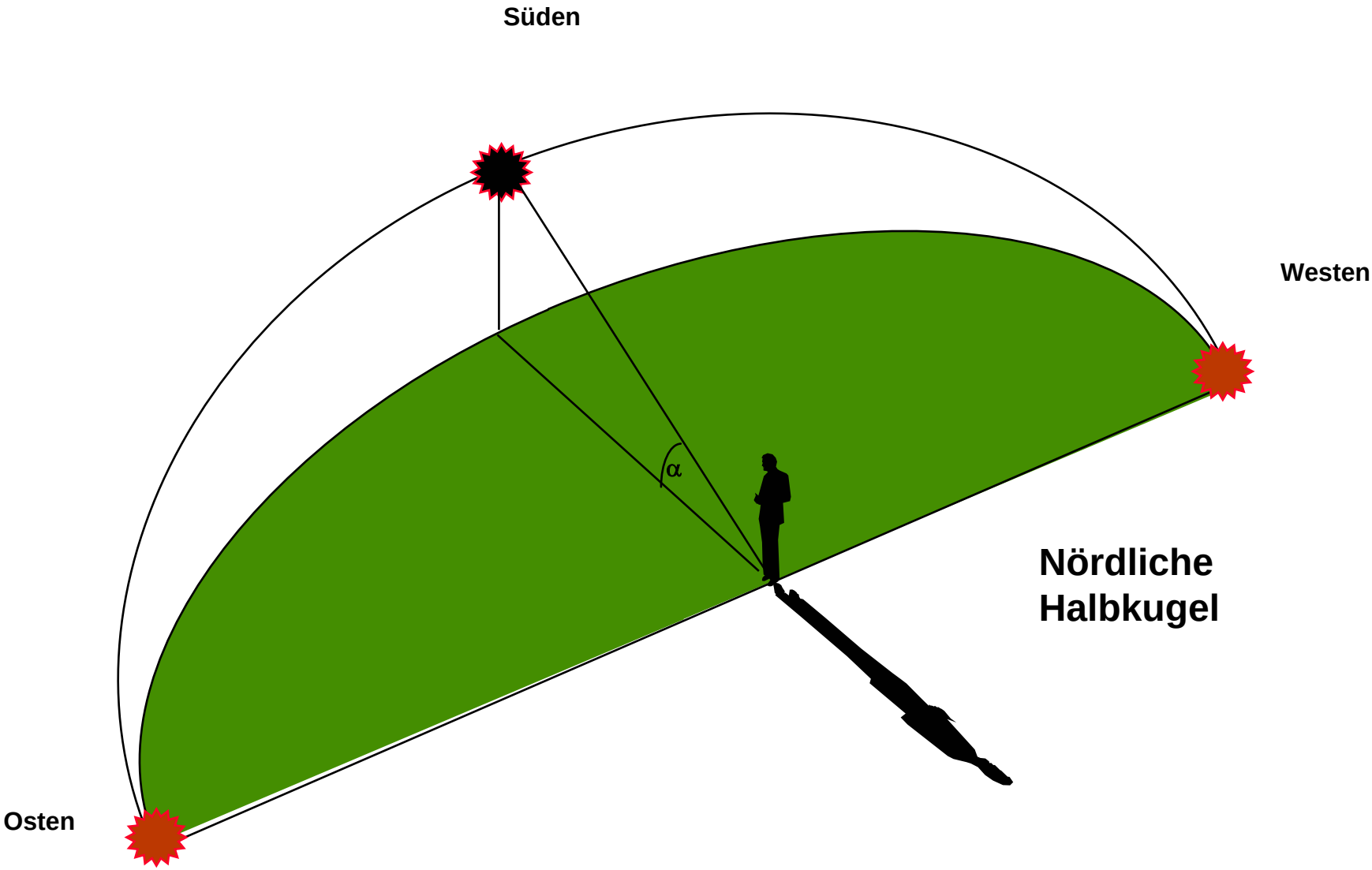
Wozu dient Zeit ?

Orientierung:
Einordnung in den kontinuierlichen „Zeitstrom“

Regulierung
Durchsetzung einer „Zeitdisziplin“

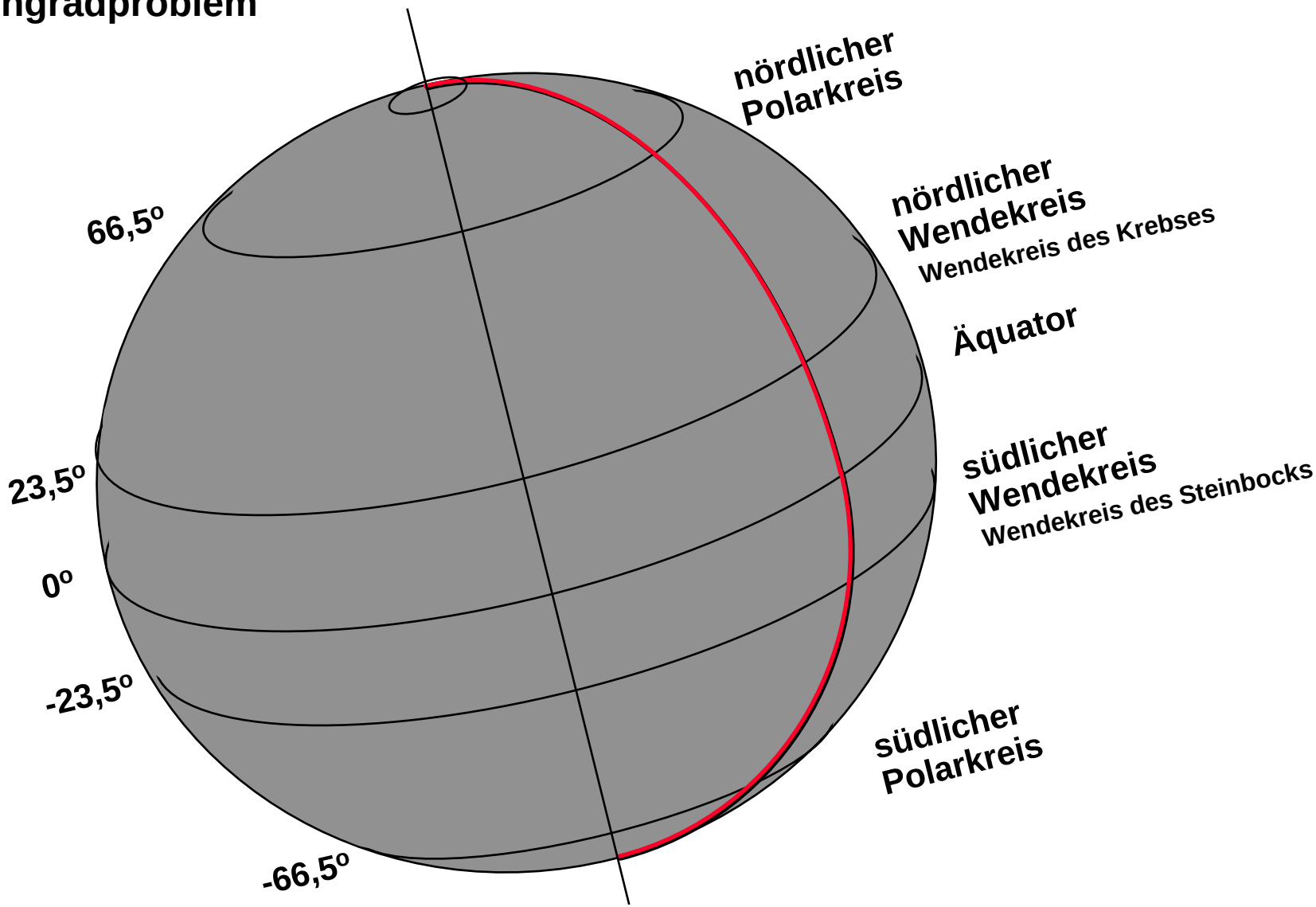
Koordinierung
Abstimmung zur Durchführung gemeinsamer Aufgaben

Wie wird die Zeit bestimmt?



Aus der Höhe (Elevation) der Sonne über dem Horizont läßt sich anhand eines Kalenders die geographische Breite ermitteln. Die höchste Elevation hat die Sonne um 12.00 Uhr Ortszeit.

Das Längengradproblem



Bestimmung der Ortszeit durch Zenit-Teleskop

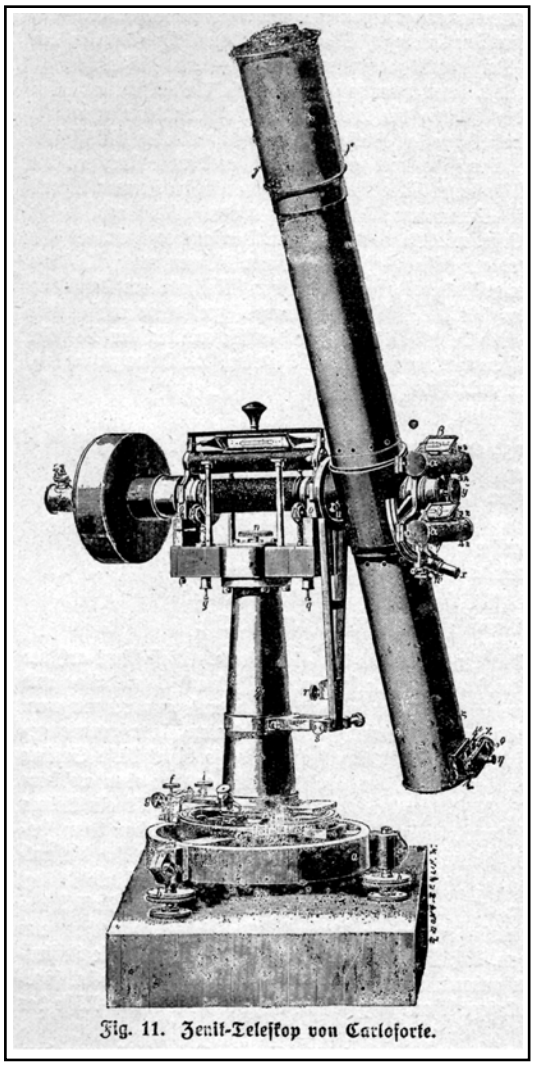


Fig. 11. Zenit-Teleskop von Carloforte.

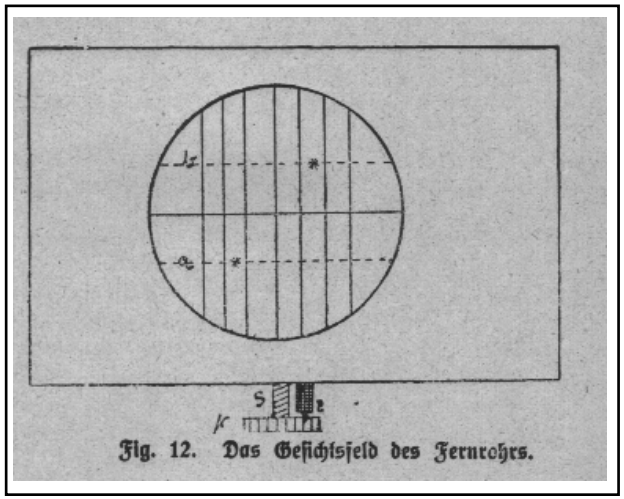
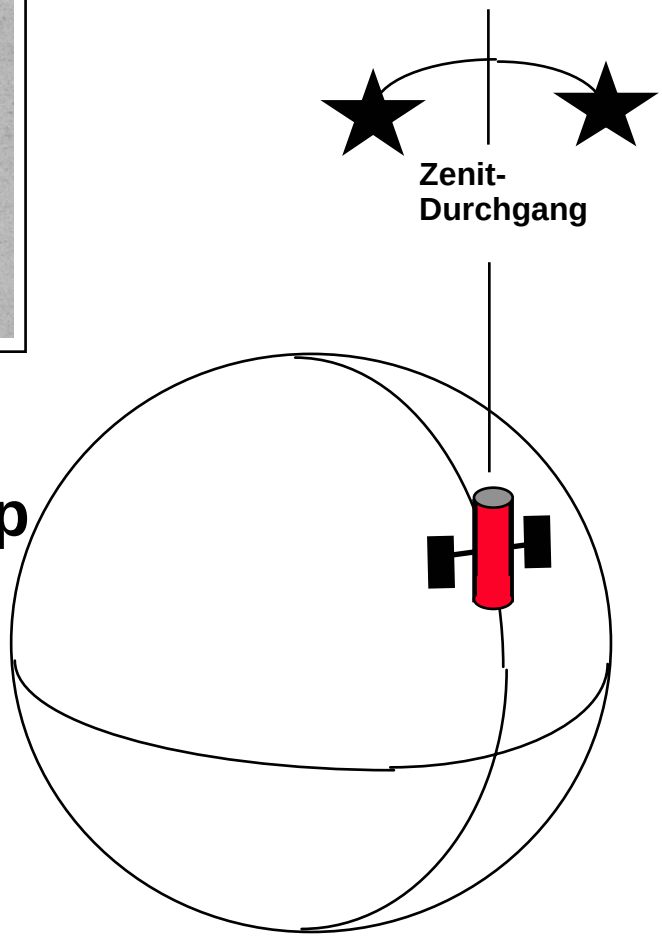


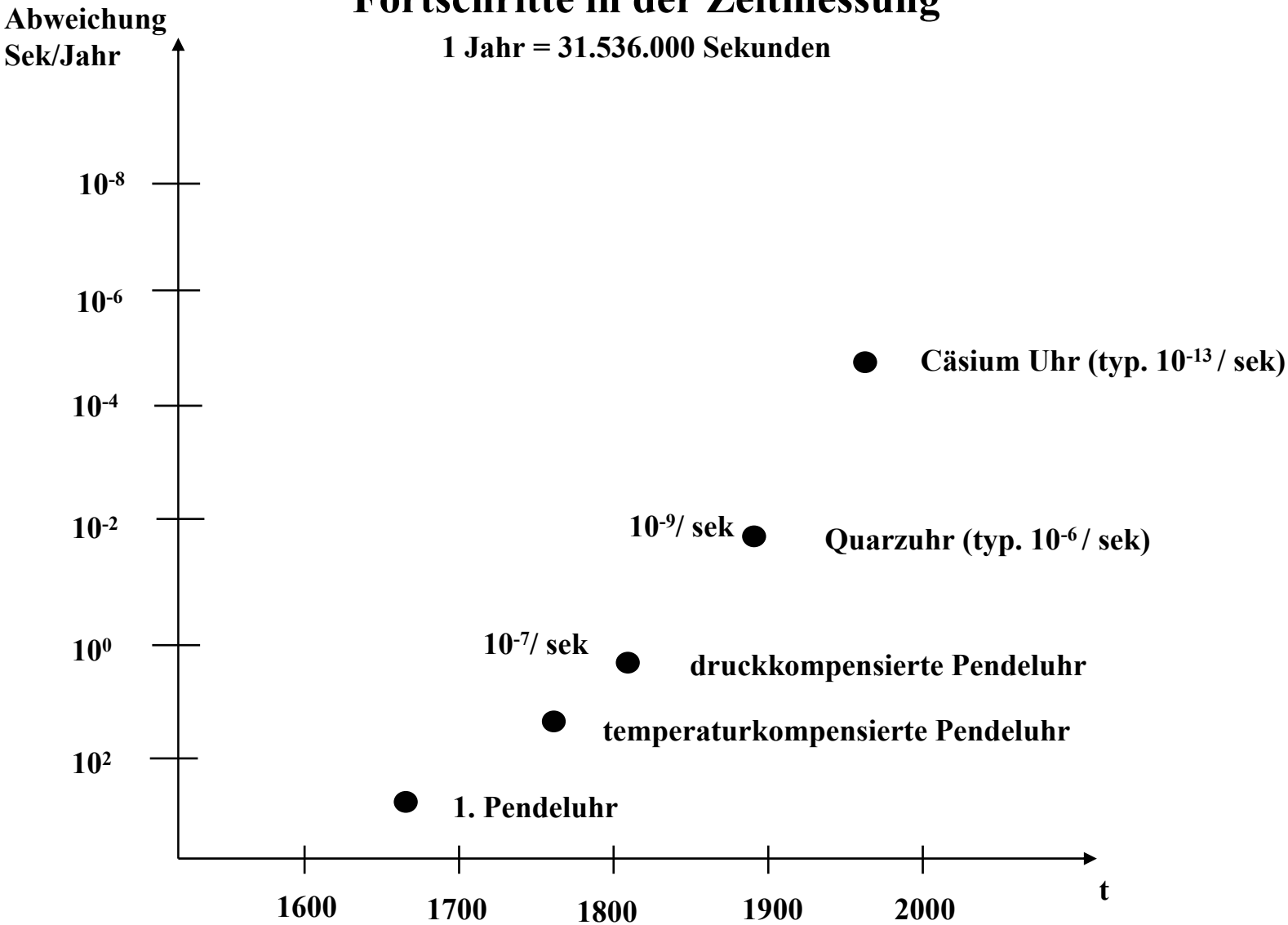
Fig. 12. Das Gesichtsfeld des Fernrohrs.

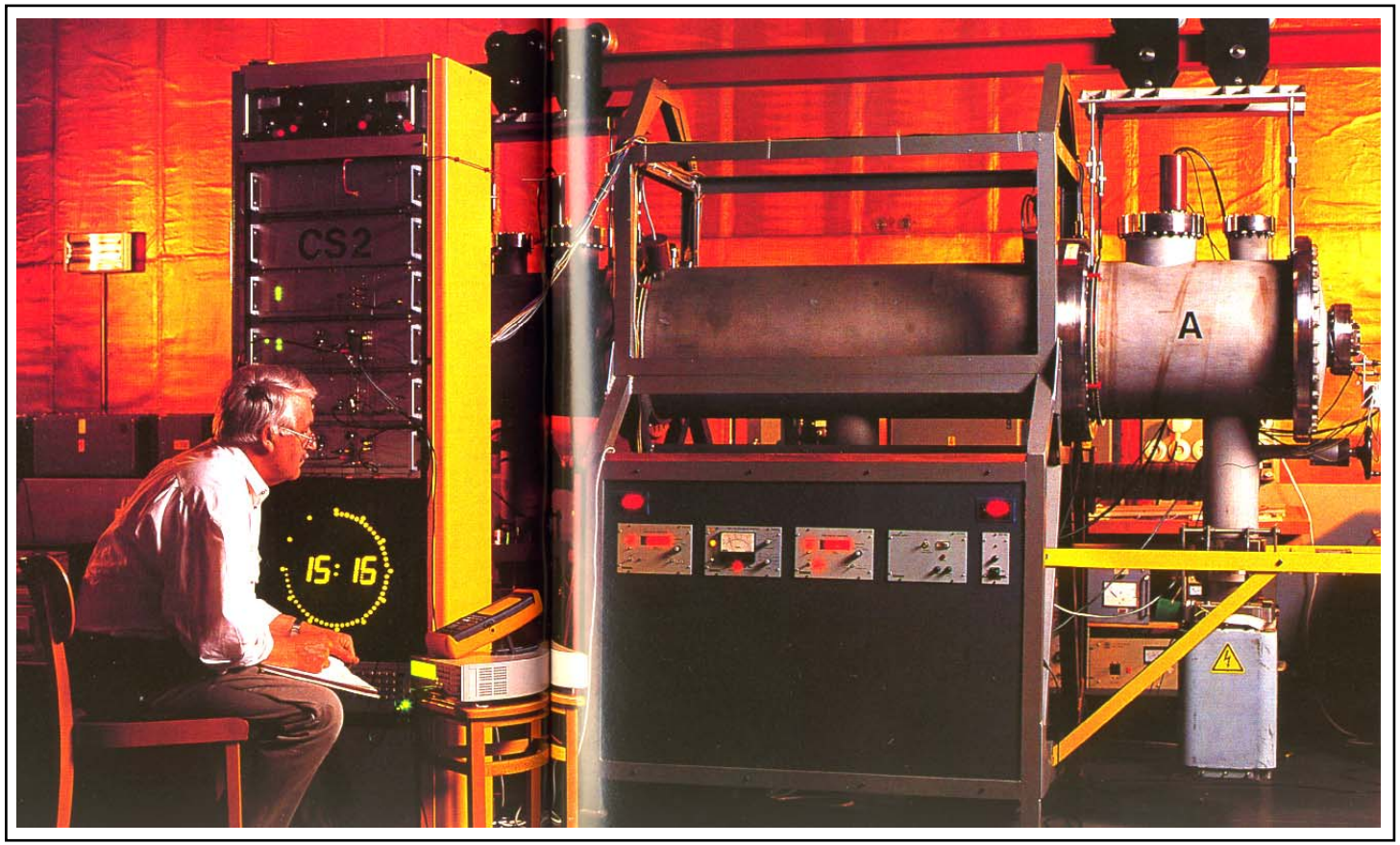
Das Zenit-Teleskop



Fortschritte in der Zeitmessung

1 Jahr = 31.536.000 Sekunden





Die Cäsium-Uhr der Physikalisch-Technischen Bundesanstalt in Braunschweig

D. Lehmann: "Ohne Uhren keine Zeit", in: Geo - Das neue Gesicht der Erde,
Nr.12, Dezember 1995

Zeitstandards

Astronomische Zeit (AT)		**basiert auf der gleichförmigen Bewegung von Himmelskörpern**
Physikalische Zeit (PT)		**basiert auf periodischen physikalischen Prozessen (Schwingungen)**
Sonnendurchlauf:		**Höchster Punkt der Sonne während eines Tages**
Sonnentag:		**Intervall zwischen zwei aufeinanderfolgenden Sonnendurchläufen**
Sonnensekunde:		**Der 1 / 86400 Teil eines Sonnentages**
Mittlere Sonnensekunde:		**Sonnensekunde gemittelt über eine große Anzahl von Sonnentagen Erster Zeitstandard (1820) basiert auf mittlerer Sonnensekunde**
Zeitzonen:		**Gebiete für die dieselbe Zeit festgelegt ist. 1884 wird die Welt in 24 Zeitzonen aufgeteilt. Die Zeitzonen unterscheiden sich von UT (GMT) ganzzahlig um jeweils 1 Stunde.**
UT	**(AT, 1833)**	**Universal Time (UT) Mittlere Sonnenzeit, gemessen am Greenwich 0-Meridian (GMT). Basiert auf der mittleren Länge eines Sonnentags, d.h. auf der Erdrotation**
ET	**(AT, 1955)**	**Ephimeridenzeit (ET), basiert auf der Umlaufzeit der Erde um die Sonne. Harold Spencer Jones stellte 1939 fest, daß die Rotation der Erde variiert, die Umlaufzeit um die Sonne nicht. 1 Sekunde der ET wird festgelegt als der 1/31.566.925,9747 Teil des tropische Jahres, das am Mittag des 1. Januars 1900 begann. (Tropische Jahr: Periode zwischen zwei aufeinaderfolgenden Durchläufen der Sonne durch den Himmelsäquator in derselben Richtung.)**
UT0	**(AT, 1960)**	**Zeit, basierend auf den lokalen Beobachtungen verschiedener, über die Erde verteilter Observatorien.**
UT1	**(AT, 1960)**	**Zeit, basierend auf der Koordination der verschiedenen UT0-Zeiten (Mittlung).**
UT2	**(AT, 1960)**	**Nochmals, auf empirischer Basis korrigierte UT1-Zeit.**
TAI	**(PT, 1961)**	**Temps Atomique International (TAI) basiert auf mehreren koordinierten Cäsium-Uhren. Fortlaufende Zeitzählung, beginnend mit dem 1. Januar 1958 0 Uhr UT2-Zeit (daher konsistent mit UT2).**
UTC	**(PT, 1972)**	**Universal Time Coordinated (UTC) basiert auf TAI, wird aber ständig an UT2 angepaßt. Immer wenn UTC und UT2 mehr als 800 ms auseinander gedrifted sind, wird eine "Schaltsekunde" eingefügt. UTC beginnt am 1. Januar 1972. Seit dieser Zeit sind (bis 1992) 15 Schaltsekunden eingefügt worden. UTC ist damit eine an AT angepaßte physikalische Zeit. Genauigkeit: ca. 1 Sek / 300000 Jahre**

Weltweite, standarisierte Zeitreferenz: *Universal Coordinated Time* (UTC)

verteilt durch
- **landgestützte Radiosender (0,1-10 msek)**
- **GPS - Satellitengruppe (*Global Positioning System*)**
- **GEOS - Satellitengruppe (*Geostationary Environmental Operational Satellites*)**

Gängige Zeitbasis in verteilten Systemen:
- **eine UTC-basierte Referenzzeit**
- **lokale Uhren**
- **Synchronisationsalgorithmus**

Übereinstimmung mit perfekter Zeit hängt dann ab von
- **der Präzision des Signals**
- **dem Abstand vom Sender**
- **atmosphärischen Bedingungen**
- **Drift der lokalen Uhren**
- **Kalkulierbarkeit der Kommunikationsdauer im Netz**
- **dem Synchronisationsalgorithmus**

Global Positioning System (GPS)

- **24 Satelliten, 20.000 km Bahnhöhe, kreisförmige Umlaufbahn, Umlaufzeit 12h**
- **min 4 Satelliten sind jeweils "sichtbar"**
- **Jeder GPS-Satellit identifiziert sich durch einen Code mit:**
  Kennung, eigene Bahnparameter, Position, Uhrzeit, Bahnparameter anderer Satelliten, Zustand der Geräte an Bord.
- **Kennung wird jede 1 ms wiederholt,**
- **Sendefrequenz : 1,6 GHz**

- **Uhren: 3 Cäsium-Uhren an Bord; werden von Bodenstation gegenüber den Uhren der anderen Satelliten auf ±5ns synchronisiert.**

- **Standortbestimmung basiert auf Abstandsmessung durch Laufzeitberechnung:**
  $R = (T_o - T_s) \cdot c$ (1,5m bei Abweichung von 5ns, 300m bei 1 µs)

- **Bei stationären Empfangsstationen läßt sich durch die Ausnutzung des Dopplereffektes eine Positionsbestimmung im mm-Bereich durchführen (Geodäsie).**

- **Künstliche Ungenauigkeiten**

- **Differentielles GPS nutzt bekannte Bodenstationen zum Ausgleich.**

Physikalisches Modell der Zeit:

Voraussetzung: Existenz einer als Zeitreferenz geeigneten Ereignisfolge.

Eigenschaften:

Äquivalenz:	**Alle Zeitstempel sind äquivalent in dem Sinne, daß eine Formel zur Konvertierung von einem zum anderen existiert.**
Linearität:	**Die Konvertierungsformeln sind linear. Verschiedenen Zeitstempel können mit Hilfe eines Proportionalitätsfaktors miteinander in Beziehung gesetzt werden.**
Kausalität:	**Zeit ist ausgerichtet gemäß einer Ordnung von Ereignissen, die zueinanderin einer vorher/nachher-Beziehung stehen.**
Zeitumkehr:	**Physische Prozesse sind i.a. nicht umkehrbar.**
Homogenität:	**Zeit ist gleichförmig und enthält keine Lücken.**

Beziehung zwischen interner und externer Zeit

Um das Auftreten von Ereignissen in der Metrik der externen physikalischen Zeit (TAI) zu messen, muß es eine interne Abbildung der physikalischen Zeit im Rechner geben.

Interne physikalische Zeit:	**Interne Zeitreferenz, welche die Metrik der externen physikalischen Zeit approximiert. Lokale Zeitreferenz. Alle Ereignisse sind in den globalen Zeitstrom eingereiht. Die vor- nach Relation zwischen Ereignissen ist durch den Zeitpunkt ihres Auftretens gemäß der internen Zeit bestimmt. Auch kausal unabhängige Ereignisse werden zeitlich geordnet. Gegeben zwei Ereignisse e_1 und e_2 zu den Zeitpunkten t_{e1} und t_{e2} Wenn für die Ereignisse auf einer internen Zeitscala $t_{e1} \leq t_{e2}$ gilt, dann muß dies auch auf der externen phys. Zeitskala gelten.**
Interne Synchronisation:	**Die Synchronisation der Uhren in einem verteilten System untereinander.**
Externe Synchronisation:	**Die Synchronisation der Uhren mit der externen physikalischen Zeit.**

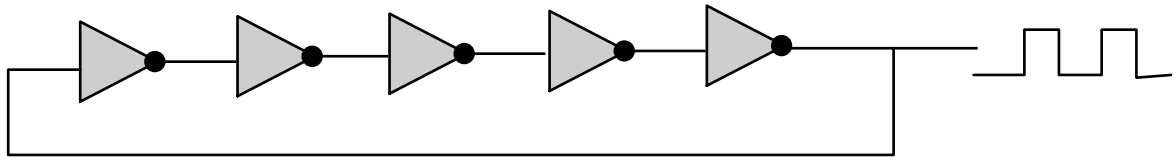
Physikalische Uhren:

Physikalische Uhr: Zeitmesser, der auf periodischen physikalischen Prozessen (Schwingungen) beruht (Pendel, Quarz, Atom), um den Fortschritt der Zeit zu messen.

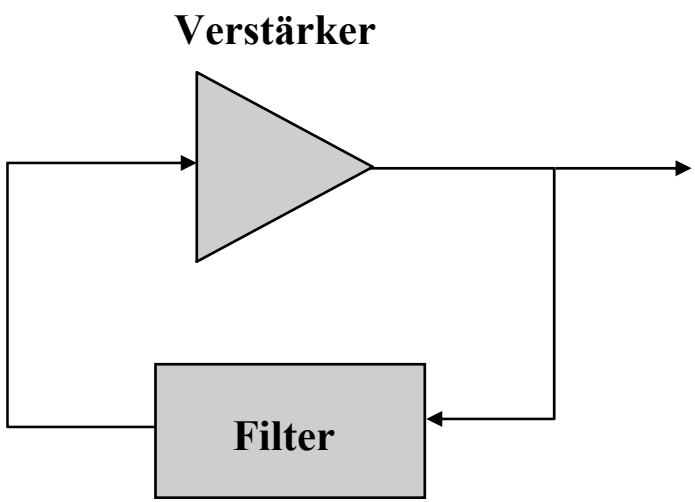
Charakterisierung physikalischer Uhren:

Frequenz:	**f**	**Anzahl der periodischen Ereignisse/Zeiteinheit (Schwingungen, Taktimpulse)** **Metrik: Hz, KHz, MHz, GHz, . . .**
Periode:	**p=1/f**	**Dauer zwischen zwei aufeinanderfolgenden periodischen Ereignissen.** **Metrik: sek, ms, µs, ns . .**
Granularität: (Auflösung)	**g**	**Zeitintervall zwischen zwei Taktimpulsen (entspricht einer Periode).** **Kann nur durch eine Uhr höherer Granularität direkt bestimmt werden.** **Bestimmt die untere Grenze des zeitlichen Abstands zweier Ereignisse, die als nicht gleichzeitig wahrgenommen werden.** **Metrik: sek, ms, µs, ns . . .**

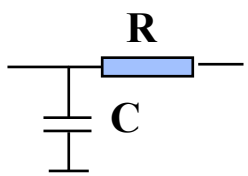
Einfacher Oszillator

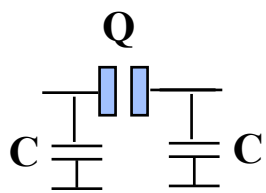


Oszillator



Alternativen, z.B.:





Sei C(t) die Funktion, welche einen Zeitpunkt t der Referenzeit auf die interne Uhr abbildet.

Def. 1:
eine Uhr heißt *Referenzuhr* , wenn für alle t gilt: C(t) = t

Def. 2:
eine Uhr ist korrekt zu einem Zeitpunkt t_0 , wenn gilt: $C(t_0) = t_0$

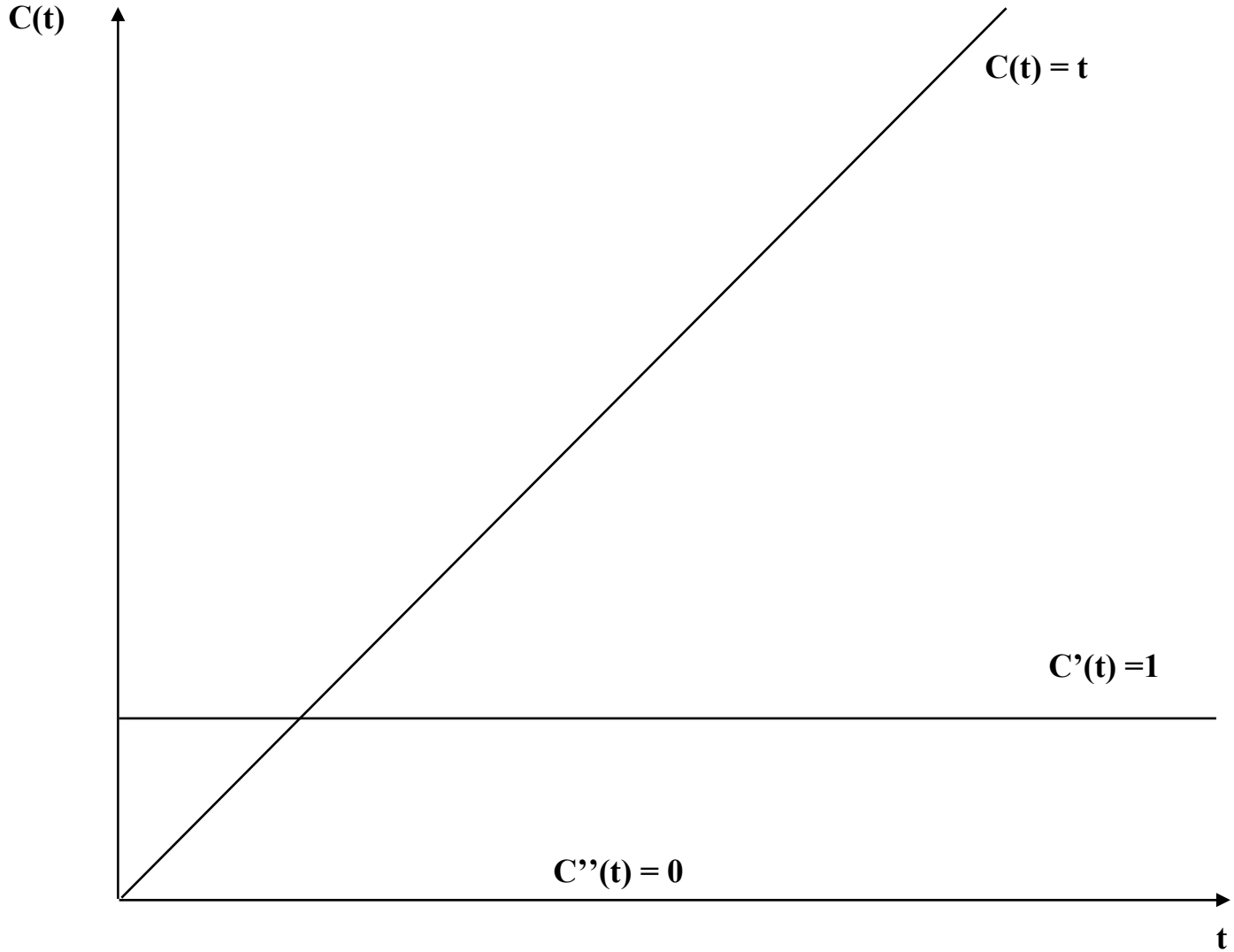
Def. 3:
eine Uhr geht genau zum Zeitpunkt t_0 , wenn an t_0 gilt: $C'(t) = 1$ ($dC(t) / dt = 1$)

C'(t) gibt die *Ganggenauigkeit* einer Uhr an.
Man kann 3 Fälle unterscheiden:
dC/dt > 1: Uhr geht zu schnell,
dC/dt = 1: Uhr geht genau,
dC/dt < 1: Uhr geht zu langsam

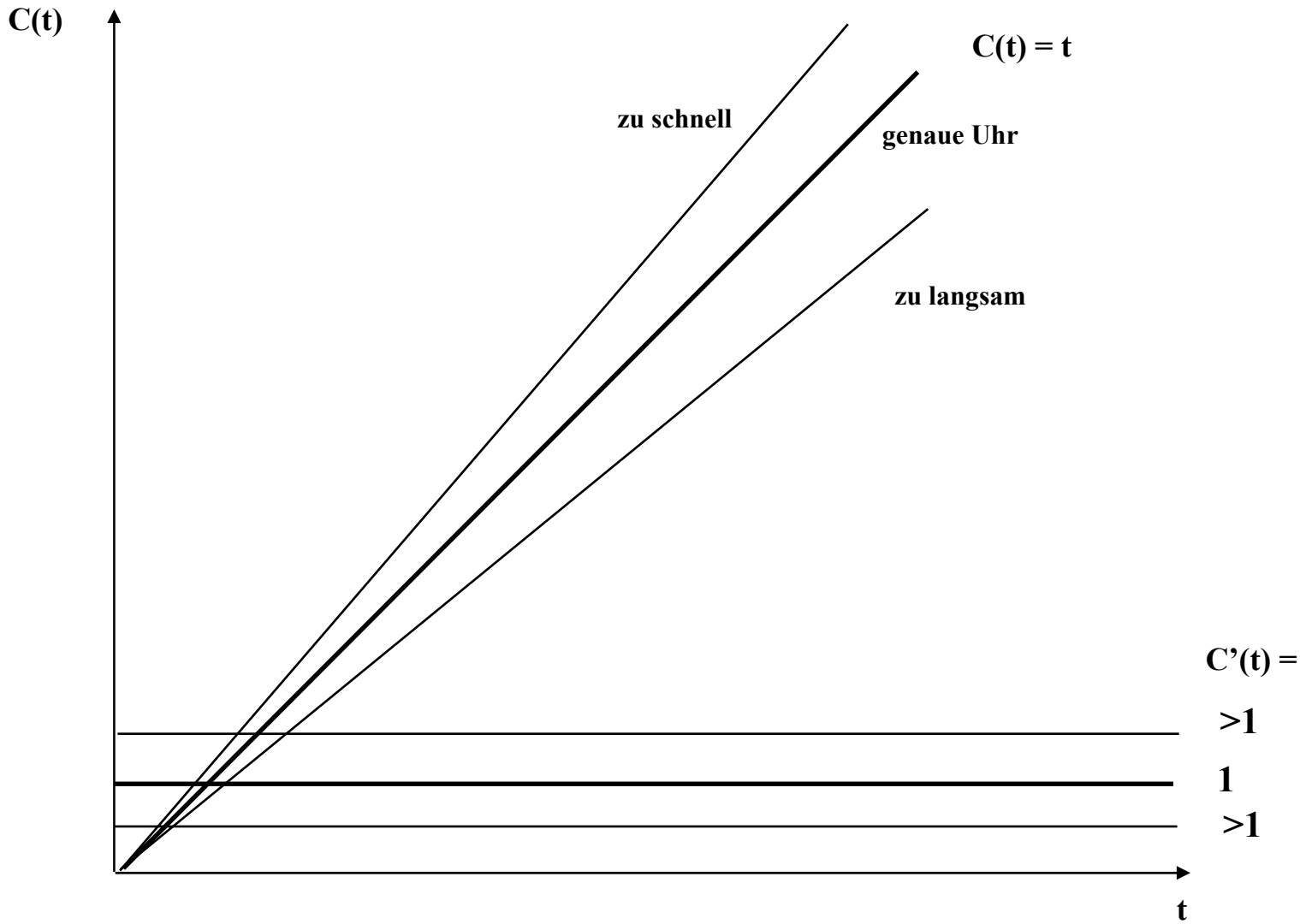
Drift(rate) r : Abweichung der Uhr / Zeit

$1-\rho \leq dC/dt \leq 1+\rho$

Die ideale Uhr

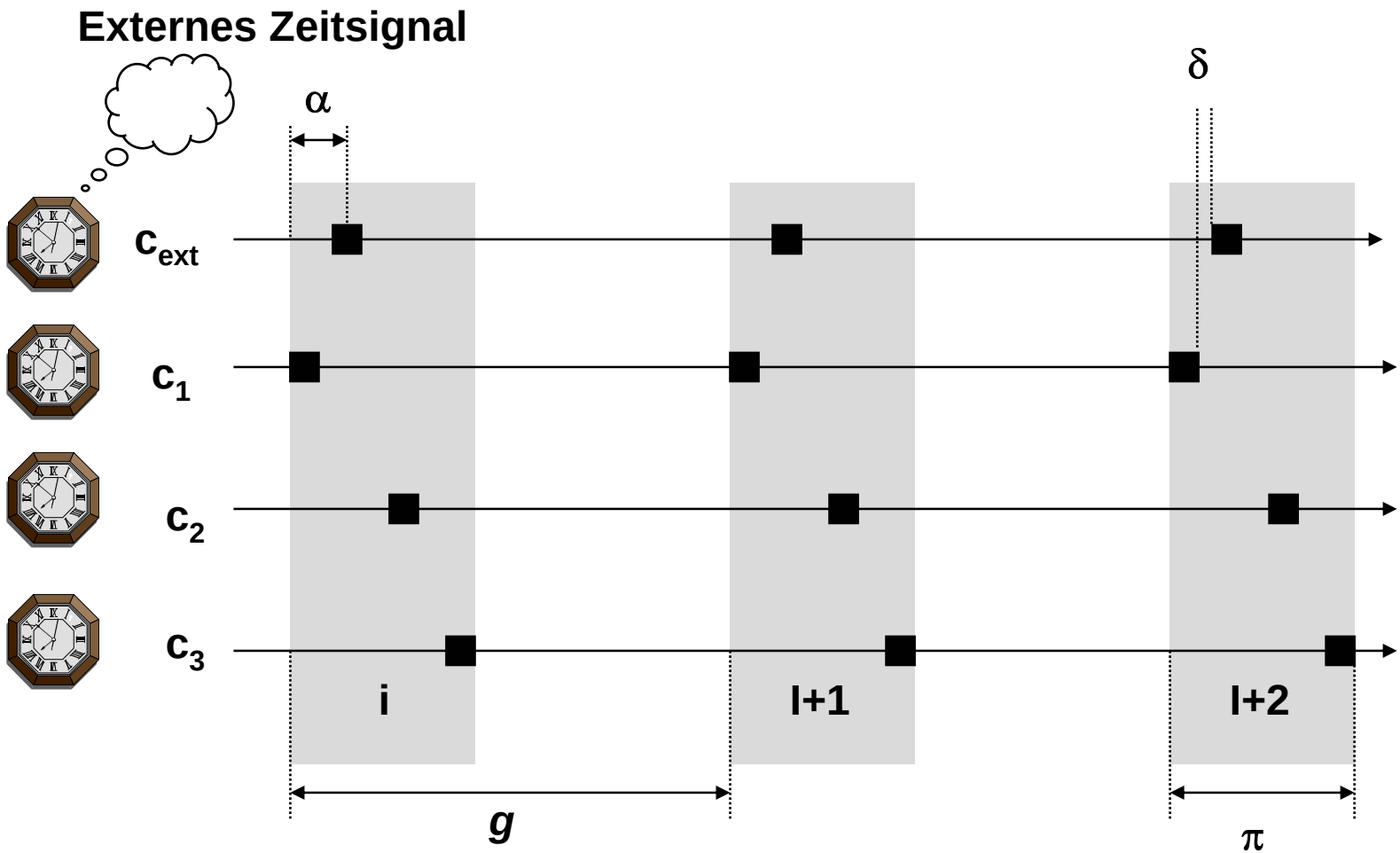


Uhr mit konstanter Drift



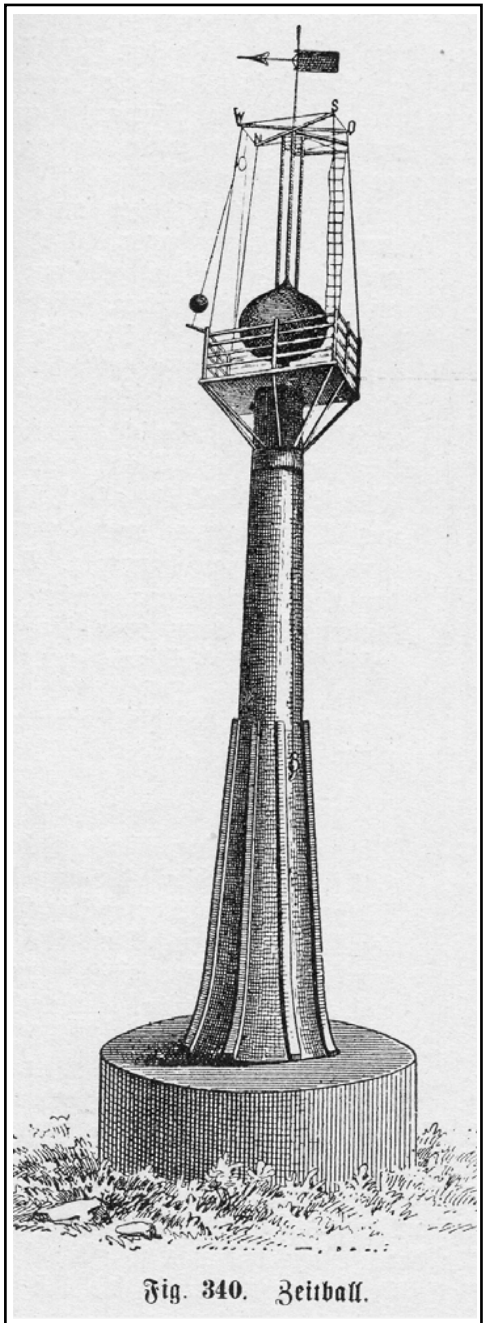
Terminologie

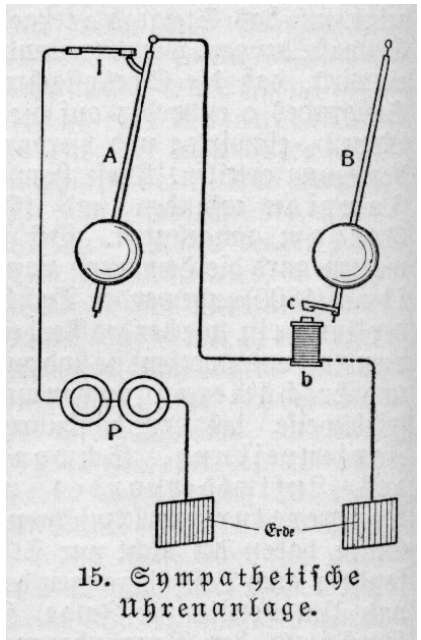
Ganggenauigkeit, (stability)	**:**	**Abweichung von idealer Frequenz**
Zeitgenauigkeit α (accuracy)	**:**	**Abweichung von einem Zeitnormal**
Granularität *g* (granularity)	**:**	**Kleinste meßbare Zeitdifferenz**
Gleichlauf π (precision)	**:**	**Abstand zweier Uhren voneínander**
Konvergenz	**:**	**Abstand zweier Uhren voneinander sofort nach der Synchronisation**
Offset δ	**:**	**Zeitdifferenz zwischen zwei Uhren**
Skew	**:**	**Frequenzdifferenz zwischen zwei Oszillatoren**
Drift ρ	**:**	**Langsames Entfernen von einer Referenzzeit**

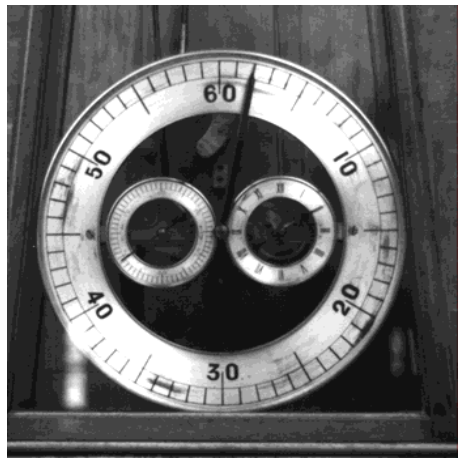


α : accuracy (Zeitgenauigkeit)
π : precision (Gleichlauf)
g : granularity (Granularität)
δ: offset (Differenz zweier Uhren)

Uhrensynchronisation







Direkte elektromagnetische Synchronisation

Zeitsignal zur radiogestützten Synchronisation

Synchronizität von Uhren

Zwei Uhren c_i und c_j sind synchron zur Uhrzeit T, wenn gilt:

$$| c_i(T) - c_j(T)| < \delta$$

Maßnahme	Ziel
Synchronisation der Frequenz	**Ganggenauigkeit**
Synchronisation der Zeit	**Zeitgenauigkeit**

Quantifizierung der Drift: Wann Resynchronisation?

Maximale Driftrate ρ:

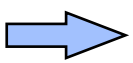
$$1-\rho \leq \frac{dC}{dt} \leq 1+\rho$$

Konstante ρ gehört zur Uhren-Spezifikation; typisch: 10^{-5} bis 10^{-6}

Abstand zweier synchronisierter Uhren nach Δt:

$$\leq 2\rho\Delta t$$

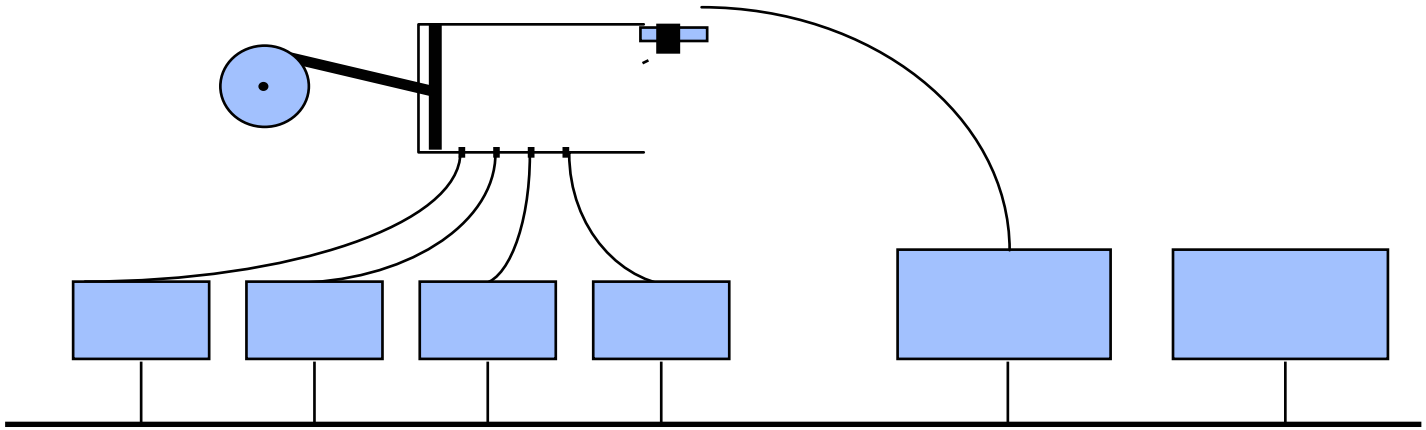
Resynchronisation zweier Uhren, die nicht weiter als δ abweichen sollen, nach



δ / 2ρ Sekunden

Drift physikalischer Rechneruhren

Driftraten heutiger Uhren 10^{-5} - 10^{-6} (~1sek in 1 - 11 Tagen).

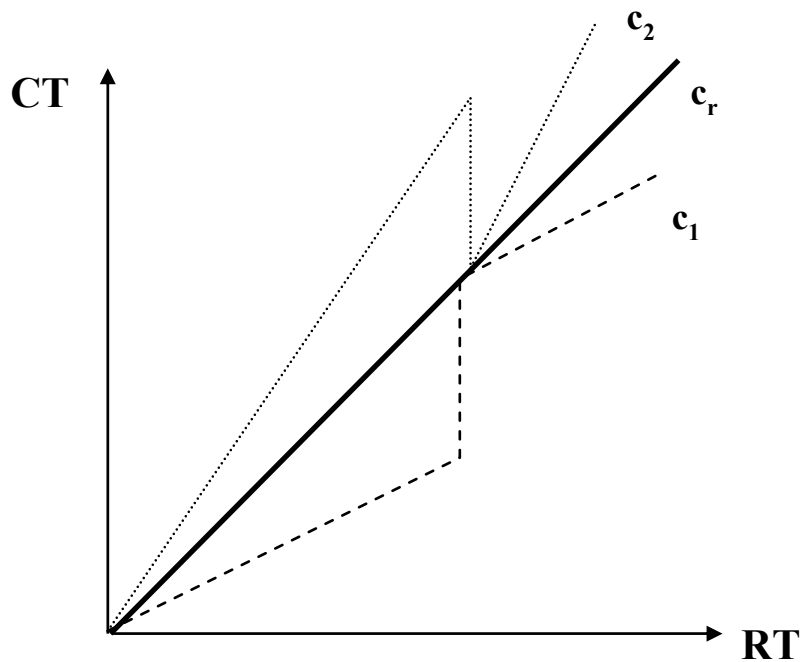


Uhr mit einer Frequenz von 10 Mhz; Periode = 100ns
Nach 10 Sek Versuchsdauer ist der worst-case Abstand zwischen zwei Uhren:

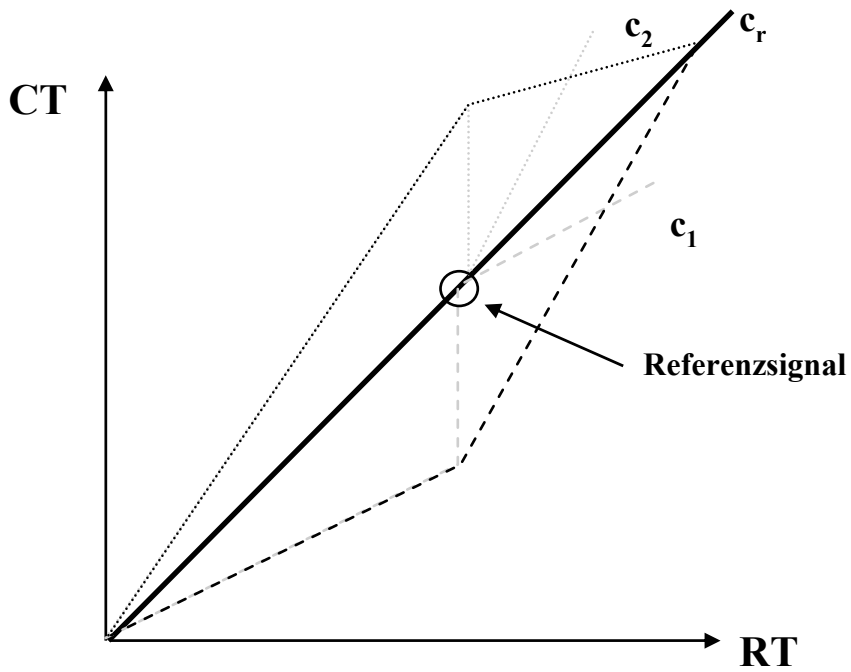
$10*2*10^{-6} = 2*10^{-5} = 20\mu sek = 20.000$ nsek

⇨ 200-fache Auflösungsvermögen der Uhr!!

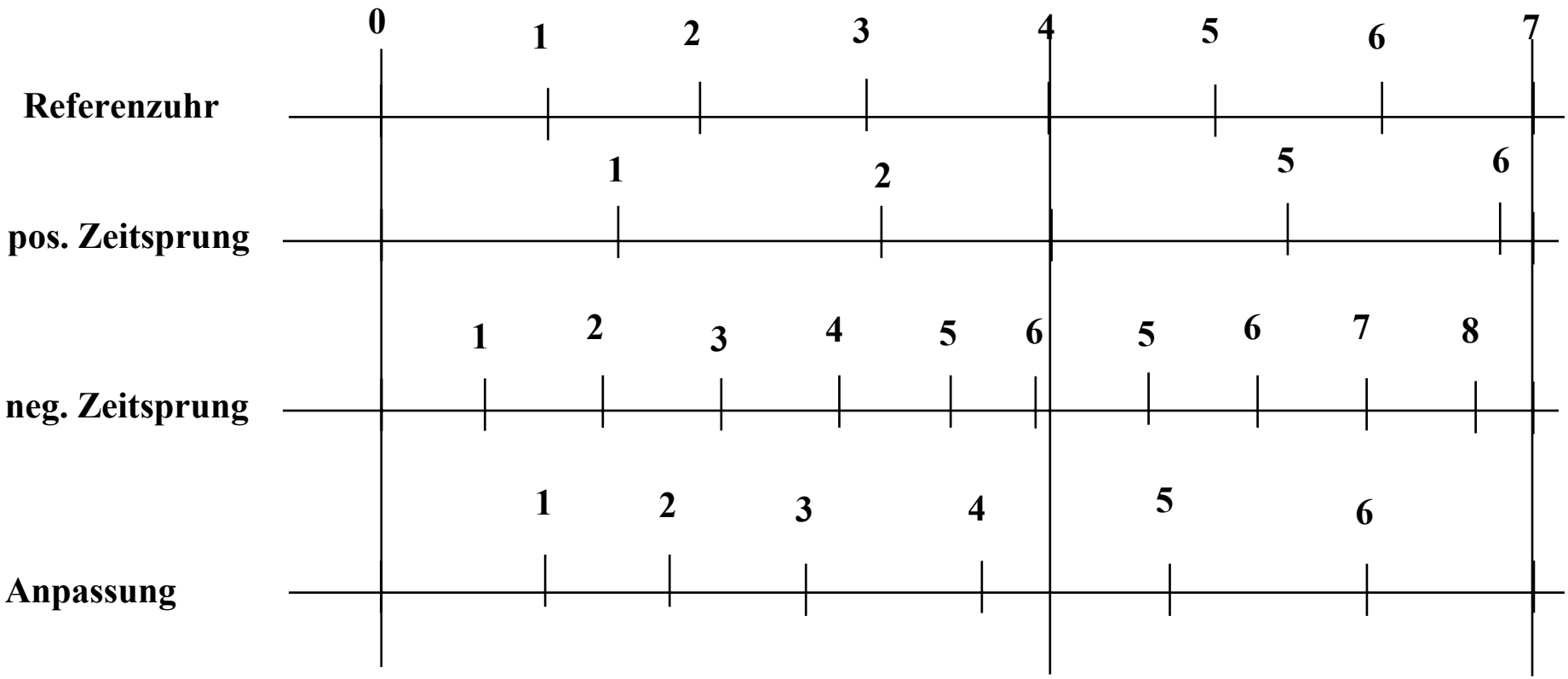
Wie Synchronisation ?



Wie Synchronisation ?

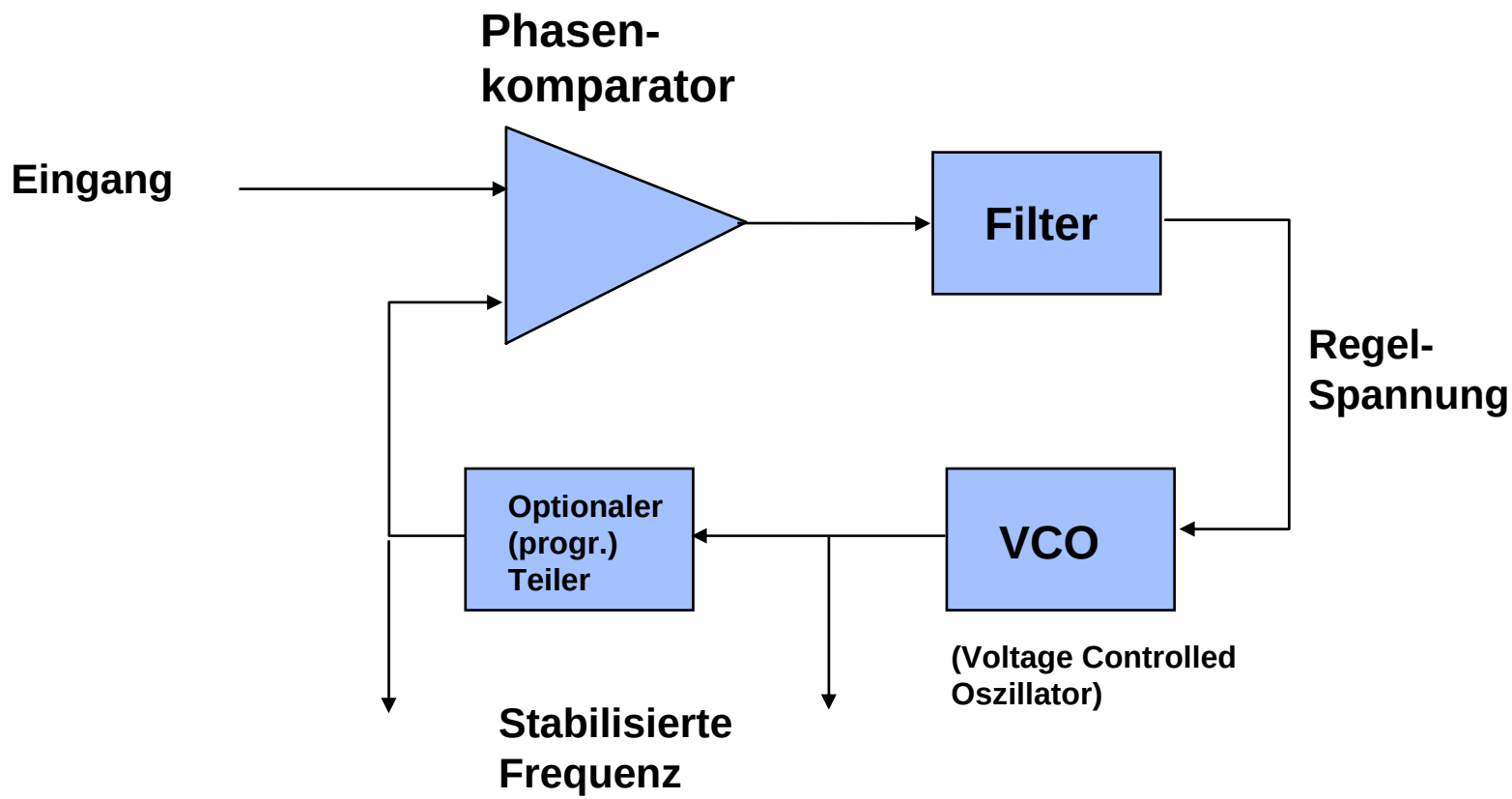


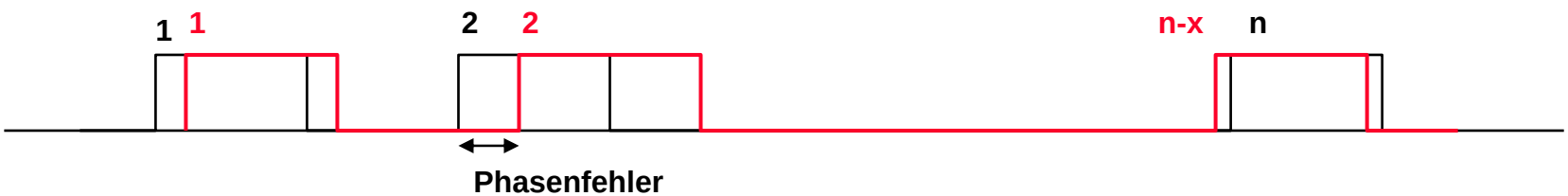
Korrektur des Zählerstandes
Korrektur der Drift



Uhrensynchronisation über spezielle Leitungen/Netzwerke auf der physischen Ebene

PLL: Phase Locked Loop
Phasenverriegelte Kontrollschleife





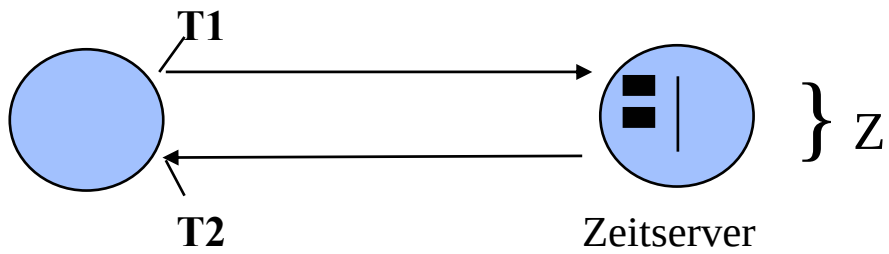
Uhrensynchronisation über Nachrichten in allgemeinen Kommunikationsnetzen

- Zeitserver
- Dezentrale und kooperative Verfahren

Uhrensynchronisation durch Master/Slave Verfahren (Master = Zeitserver)

Algorithmus nach Cristian:
Jeder Rechner resynchronisiert periodisch seine Zeitbasis mit dem Zeitserver

(nach $\frac{\delta}{2\rho}$ Sekunden)

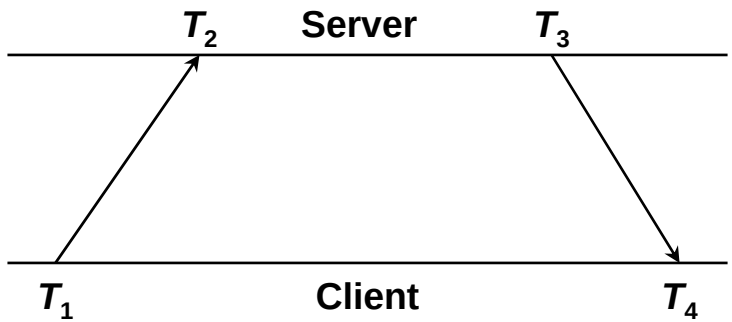


(a) Anfordern der Zeit beim Zeitserver
(b) Korrektur des erhaltenen Wertes um Korrekturfaktor **K**

Probleme:
- Monotonieeigenschaft der Zeit
- Messen eines “round trips”; variiert!
  -> einfaches Verfahren: Korrektur um halbe round-trip - Zeit der Zeitanfrage selbst
  -> Verfeinerung: statistische Mittelwerte für T2-T1
  -> Verfeinerung: Laufzeit des Zeitservers

K = (T2 - T1 - Z) / 2

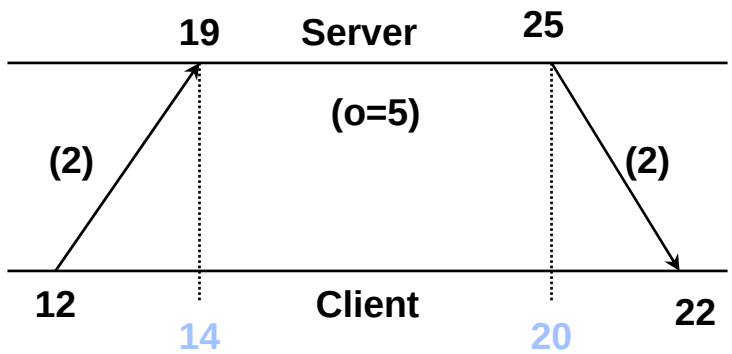
Berechnung des Offsets o und der Verzögerung d



$$o = [(T_2 - T_1) + (T_3 - T_4)] / 2$$

$$d = (T_4 - T_1) - (T_3 - T_2)$$

Berechnung des Offsets o und der Verzögerung d



o = [(19 – 12) + (25 – 22)] / 2
o = 5

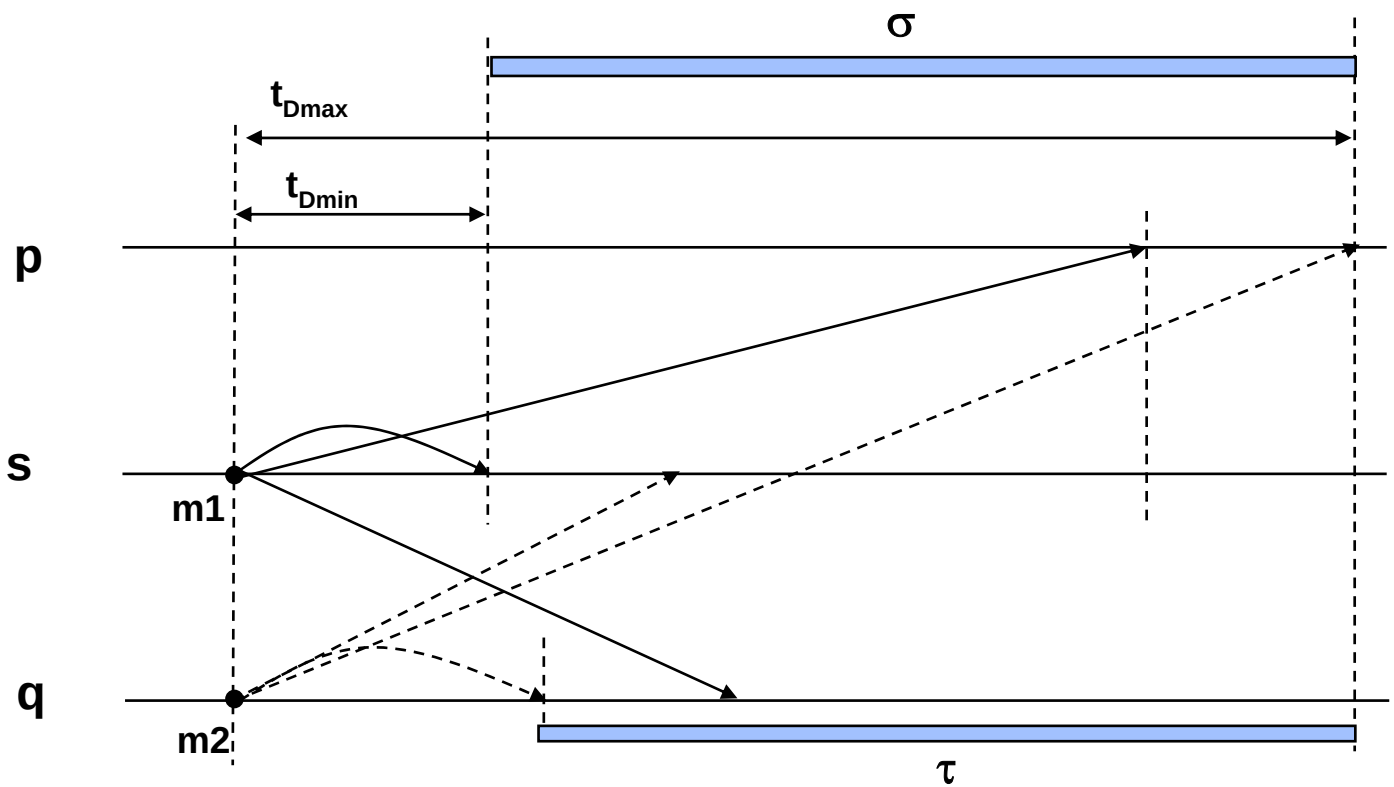
d = (22– 12) - (25 – 19)
d = 4

Parameter bei der Berechnung der Zeit:

- **Nachrichtenlaufzeiten auf dem Kommunikationsnetzwerk (steadyness + tightness)**
- **Zeit bis Nachricht verschickt wird**
  Verzögerungen in der Reply Queue
- **Lokale Auslesefehler der Uhren**
  Verzögerungen in der Request Queue
  Zeitabweichungen
  Zugriffsverzögerungen
- **Physische Parameter der Serveruhr:**
  Drift, Offset

Zur Erinnerung:

Steadyness σ : Unterschied der Laufzeiten verschiedener Nachrichten (zu versch. Knoten).
Tightness τ : Unterschied der Laufzeiten einer Nachrichten zu verschiedenen Knoten.



Das *Network Time Protocol* (Internet NTP) David L. Mills

Leistungen:

- Synchronisation der physischen Zeit in LANs/WANs mit UTC;
  statistische Techniken zur
  - Berücksichtigung von Varianz und Verzögerung der Botschaftenlaufzeiten
  - Berücksichtigung der Qualität der Zeit von unterschiedlichen Zeitservern

- hohe Verfügbarkeit durch redundante Server und Verbindungen

- Skalierbarkeit hinsichtlich der der Anzahl der Zeitserver, Klienten, und der Synchronisationsfrequenz (auf Internet-Charakteristika ausgelegt)

- Schutz vor zufälliger oder absichtlicher Manipulation der Zeitdienste; authentisierbare Botschaften mit Zeitinformationen

D. Mills : „Internet Time Synchronization: The Network Time Protocol“, IEEE TC on Communications, 39(10), 1991

http://www.eecis.udel.edu/~mills/ntp.htm

Das *Network Time Protocol* (Internet NTP): Architektur

Zwei Zeitservertypen:

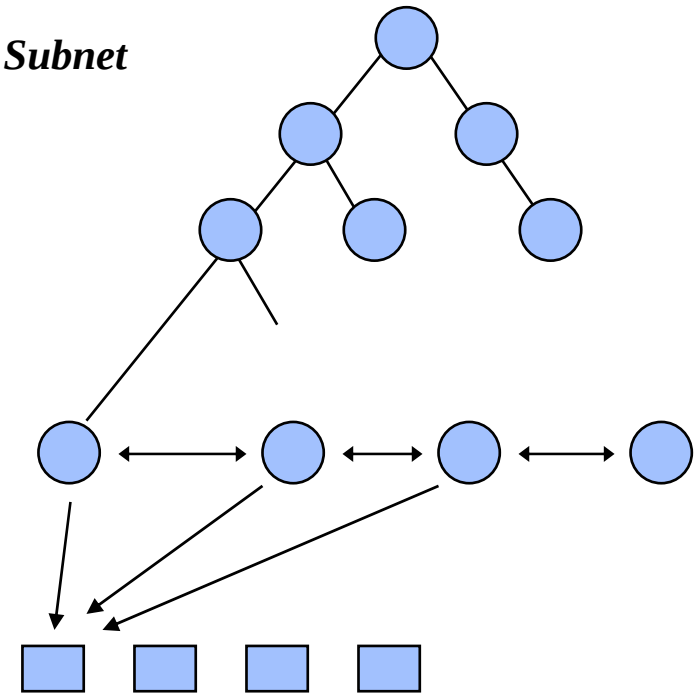
- Primärserver: UTC-Uhr
- Sekundärserver: keine UTC-Uhr, direkt oder indirekt mit einem Primärserver verbunden

Time Synchronisation Subnet

Server

Server

Server



stratum 1: Primär-Server

stratum 2: direkter Sek.-

stratum 3: indirekter Sek.-

stratum 4: zweifach-
indirekter Sek.-

• • •

Klienten

Das *Network Time Protocol* (Internet NTP): Eigenschaften

Rekonfiguration

• Ausfall der Uhr eines Primärservers: Server wird zweitklassig, zum Sekundärserver

Kommunikation

• symmetrisch: auf kleinen strati, zwischen Zeitservern in LANs; zweiseitiger Fluß von Zeitinformationen
• RPCs: Server antworten mit zeitgestempelten Botschaften, Algorithmus nach Cristian
• Multicast: durch LAN-Zeitserver

<u>Beispiel</u>: Hohe Präzision durch Kalkulation von Botschaftenlaufzeiten

Server/Server - Kommunikation: paarweise
Jeder Server fügt in jede Zeit-Botschaft **b'** ein
- den Zeitpunkt des Absendens der letzten empfangenen Botschaft **b**
- den Zeitpunkt des Empfangs der letzten empfangenen Botschaft **b**
- den Zeitpunkt des Absendens der aktuellen Botschaft **b'**

Weitere Maßnahmen zur Erhöhung der Genauigkeit

(precision + accuracy):

- **statistische Filteralgorithmen eliminieren Abweichungen von Botschaftenlaufzeiten**

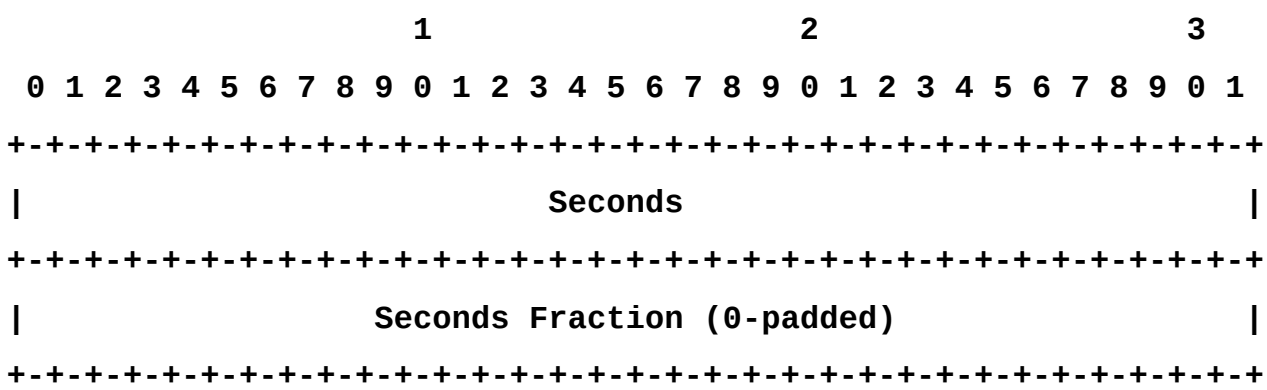
- **Gewichtung von Zeitinformationen nach ihrer Quelle (strati)**

  => Messungen: 99% aller Zeitinformationen weichen weniger als 30 msek von der realen UTC ab

Zeitformat in NTP

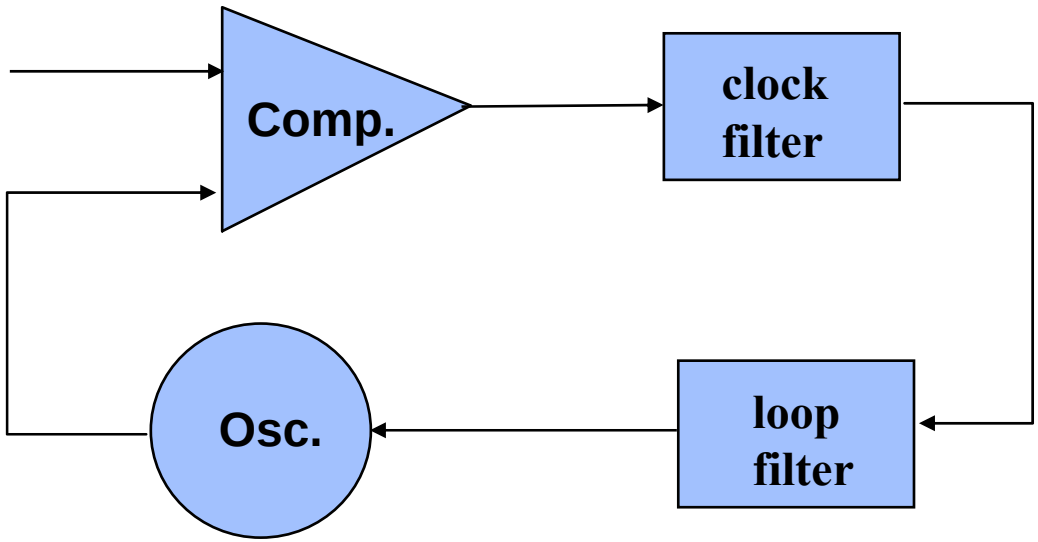
The time is the number of seconds since 00:00 (midnight) 1 January 1900

The maximum number that can be represented is
4,294,967,295 seconds with a precision of about 200 picoseconds,
which should be adequate for even the most exotic requirements.



Note that, since some time in 1968 (second 2,147,483,648) the most significant bit (bit 0 of the integer part) has been set and that the 64-bit field will overflow some time in 2036 (second 4,294,967,296).

PLL-Modell des NTP (RFC1129)

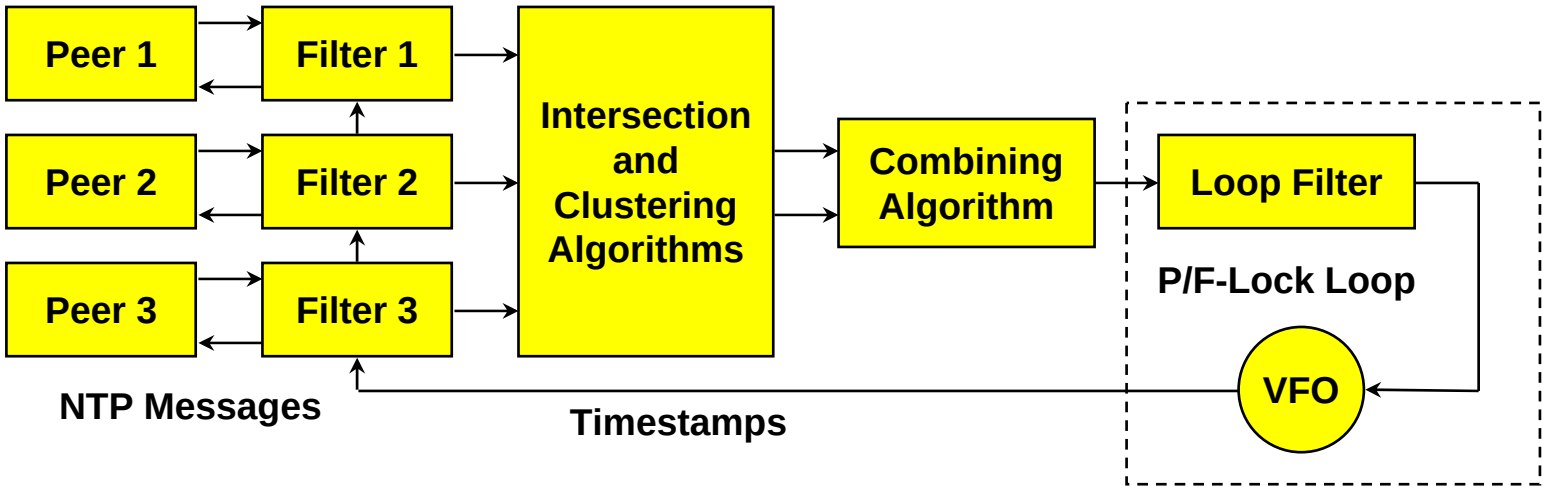


NTP reguliert die Oszillatorfrequenz

clock filter: selektiert geeignete Zeitnachrichten aus der Menge der empfangenen Zeitnachrichten

loop filter: berechnet den Korrekturwert für den Oszillator

How NTP works



- Multiple synchronization 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 servers believed to be accurate and fault-free
- Combining algorithm computes weighted average of offsets for best accuracy
- Phase/frequency-lock feedback loop disciplines local clock time and frequency to maximize accuracy and stability

D. Mills/02

Regelcharakteristik Beispiel:

Ausgangspunkt:

Phasenänderung **100ms**

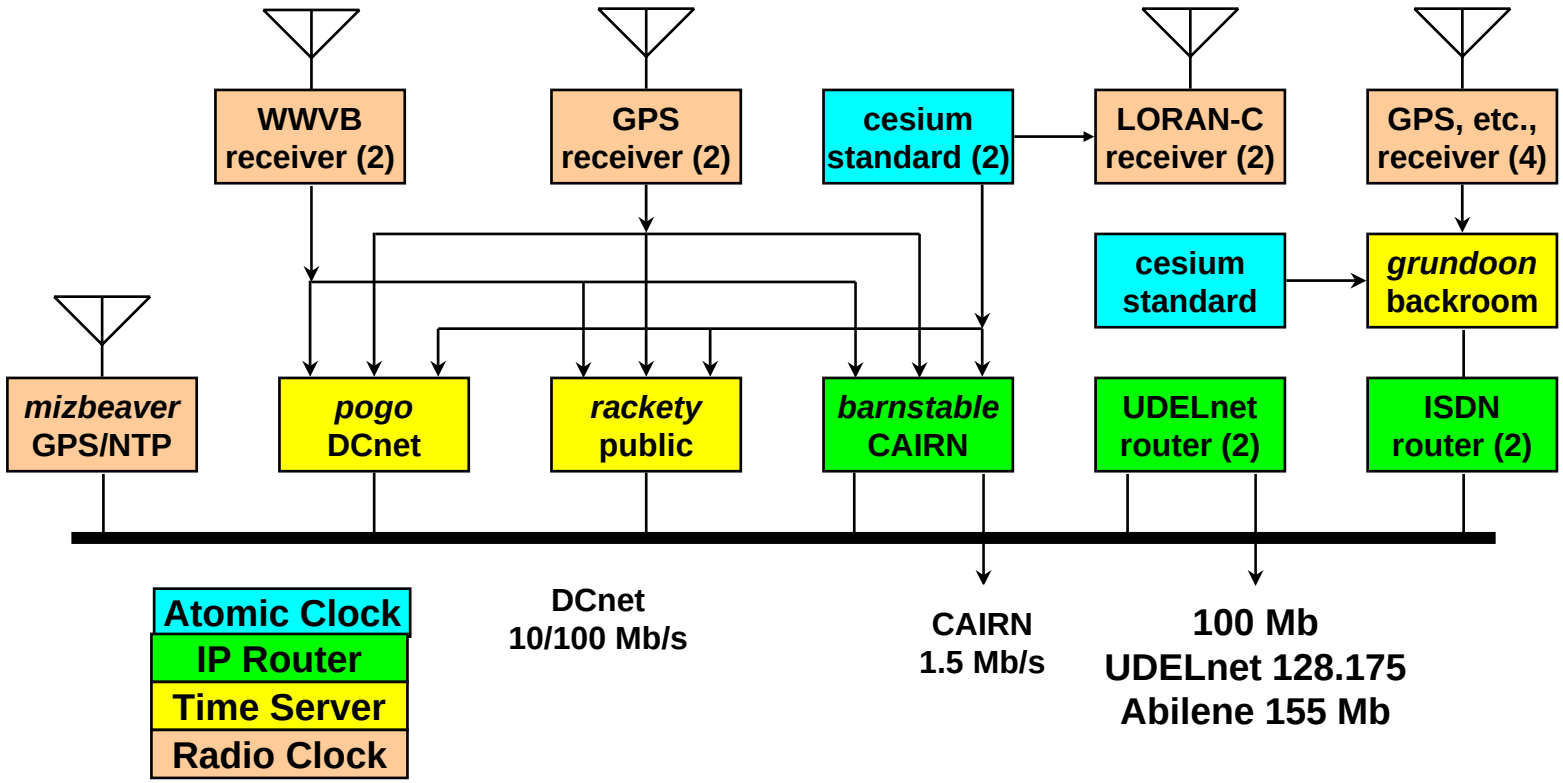
fehlerlos nach 39 Min.
Überschwingen um 7ms nach 54 Min.
Stabilisierung auf Fehler < 1ms nach 6h

Frequenzänderung **50 ppm**

1 ppm nach 16h
0,1 ppm nach 26 h

Es kann sein, dass eine Uhr außerhalb des Fangbereichs der PLL gerät. Dann wird nicht die Oszillatorfrequenz angeglichen, sondern die Uhr gesetzt.

DCnet timekeeping facilities (January 2001)



D. Mills/02

Dezentrale und kooperative Verfahren zur Uhrensynchronisation

Averaging:

L. Lamport, P. Meliar-Smith:“Synchronizing Clocks in the Presence of Faults“, Journal of the ACM, 32(1): 52-78, 1985
J. Lundelius, N. Lynch:“A new Fault-Tolerant Algorithm for Clock Synchronization“, Proc. 3rd ACM SIGACT-SIGOPS Symp. On Principles of Distr. Comp., 75-88, Vancouver, 1984

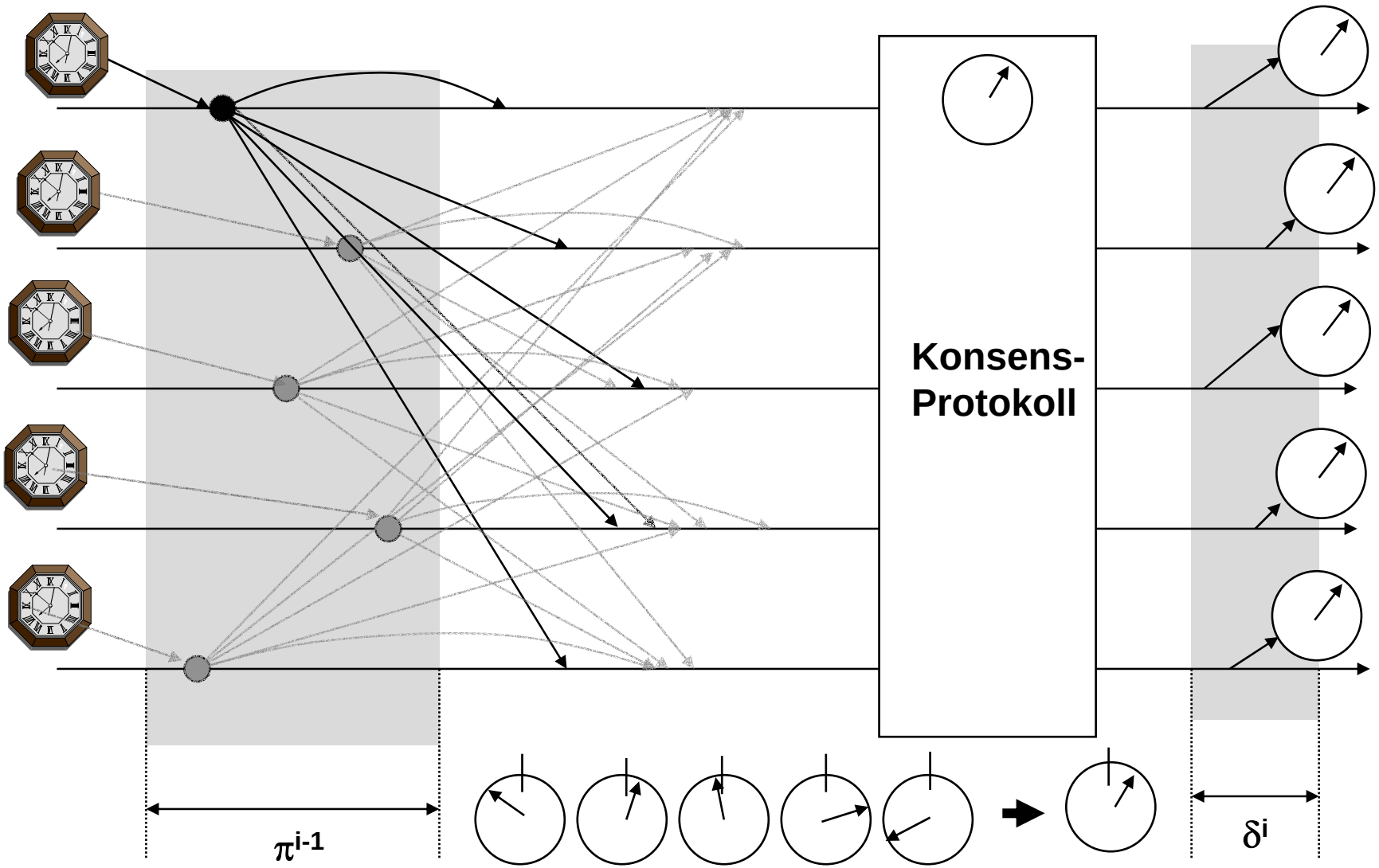
Non-averaging:

J. Halpern, B. Simons, R. Strong, D. Dolev:“Fault-Tolerant Clock Synchronization“, Proc. 3rd ACM SIGACT-SIGOPS Symp. On Principles of Distr. Comp., 89-102, Vancouver, 1984
T. Srikanth, S. Toueg:“Optimal Clock Synchronization“, Journal of the ACM,(34)3, 627-645, 1987

Hybrid:

P. Verissimo, L. Rodrigues:“A posteriori Agreement for Fault-Tolerant Clock Synchronization on Broadcast Networks“, Digest of papers, 2nd IEEE Int‘l Symp. On Fault-Tolerant Computing, Boston, 1992
M. Clegg, K. Marzullo:“ Clock Synchronization in Hard Real-Time Distributed Systems“, Technical report CS96-478, UCSD, Dept. Of Comp. Science, 1996

Mittelndes Verfahren zur Synchronisation (avaraging)



Bilden des Mittelwertes

δ : Verbesserung der „precision“ π

Kooperierende Zeitsynchronisationsverfahren (Kopetz, Ochsenreiter)*

$t^r_{i,j}$: Empfangszeit der Synchronisationsnachricht von Rechner i auf Rechner j

Sind n Rechner im Netz, so kann man die lokale Matrix der empfangenenZeitwerte konstruieren als :

*

H. Kopetz, W. Ochsenreiter
Clock Synchronisation in Distributed Real-Time Systems
IEEE Transactions on Computers, Vol. C-36, No.8
August 1988

$$C^r_j = \begin{pmatrix} t^r_{1,1} & \cdots & t^r_{1,j} & \cdots \\ \cdot & t^r_{2,2} & t^r_{2,j} & \cdots \\ \cdot & \cdot & t^r_{3,j} & \cdot \\ \cdot & \cdot & & \\ \cdot & \cdot & t^r_{j,j} & \cdot \\ \cdot & \cdot & & \cdot \\ \cdots & \cdots & t^r_{n,j} & \cdots \; t^r_{n,n} \end{pmatrix}$$

Die Diagonale enthält die Zeitstempel der Nachricht von Rechner i, die Spalte enthält dieEmpfangszeit der Nachricht auf Rechner j.

Für alle "guten" Uhren gilt:

$$t^r_{i,j} = t^r_{i,i} + (O_{i,j} + md + E^r_{i,j}) \bullet N \quad (N = 1 \text{ wenn } i \neq j,\ N = 0 \text{ wenn } i = j)$$

$O_{i,j}$ (offset) : Abweichung der Uhr auf Rechner i von der Uhr auf Rechner j
md : Verzögerung der Nachricht (angenommen)
$E^r_{i,j}$: Zusammenfassung der Auslesefehler auf Rechner i und Rechner j

Kooperierende Zeitsynchronisationsverfahren

Korrekturvektor für Rechner j

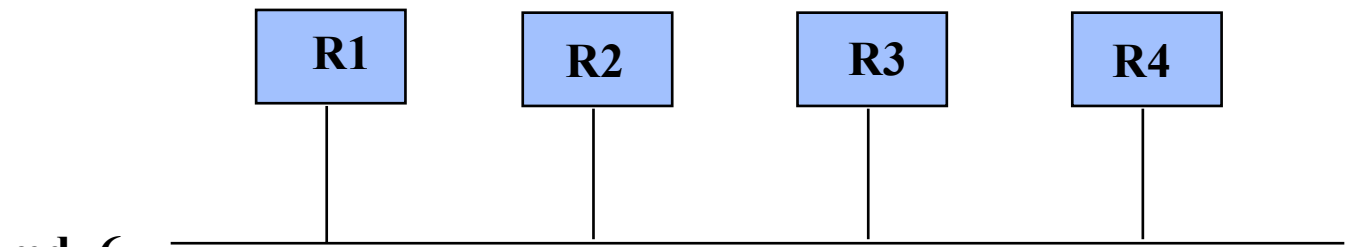
$$K^t = \begin{pmatrix} 0 & k_{1,2} & \cdots & k_{1,j} & \cdots & k_{1,n} \\ k_{2,1} & 0 & \cdots & k_{2,j} & \cdots & k_{2,n} \\ \cdot & \cdot & & k_{3,j} & & \cdot \\ \cdot & \cdot & 0 & & & \\ \cdot & \cdot & \cdots & 0 & \cdots & \cdot \\ \cdot & \cdot & & & & \cdot \\ & & & & 0 & \\ k_{n,1} & k_{n,2} & \cdots & k_{n,j} & \cdots & 0 \end{pmatrix}$$

Für jedes Element $k_{i,j}$ gilt:

$$k_{i,j} = t^r_{i,j} - t^r_{i,i} - md \bullet N \quad (N = 1 \text{ wenn } i \neq j,\ N = 0 \text{ wenn } i = j)$$
$$= (S_{i,j} + E^r_{i,j}) \bullet N$$

Kooperierende Zeitsynchronisationsverfahren

Beispiel:



$t_{1,1} = 32, 32$	$t_{1,2} = 37, 32$	$t_{1,3} = 38, 32$	$t_{1,4} = 41, 32$
$t_{2,1} = 69, 60$	$t_{2,2} = 60, 60$	$t_{2,3} = 67, 60$	$t_{2,4} = 66, 60$
$t_{3,1} = 95, 87$	$t_{3,2} = 94, 87$	$t_{3,3} = 87, 87$	$t_{3,4} = 98, 87$
$t_{4,1} = 20, 16$	$t_{4,2} = 19, 16$	$t_{4,3} = 21, 16$	$t_{4,4} = 16, 16$

$C_j^r =$

32 . . .	32 37 . .	32 . 38 .	32 . . 41
69 60 . .	. 60 . .	. 60 67 .	. 60 . 66
95 . 87 .	. 94 87 .	. . 87 .	. . 87 98
20 . . 16	. 19 . 16	. . 21 16	. . . 16
j=1	j=2	j=3	j=4

Kooperierende Zeitsynchronisationsverfahren

$C_j^r =$

32	.	.	.
69	60	.	.
95	.	87	.
20	.	.	16

32	38	.	.
.	60	.	.
.	94	87	.
.	19	.	16

32	.	38	.
.	60	67	.
.	.	87	.
.	.	21	16

32	.	.	41
.	60	.	71
.	.	87	98
.	.	.	16

j = 1

$k_{1,1} = 0$
$k_{2,1} = t_{2,1} - t_{2,2} - md = 69 - 60 - 6 = 3$
$k_{3,1} = t_{3,1} - t_{3,3} - md = 95 - 87 - 6 = 2$
$k_{4,1} = t_{4,1} - t_{4,4} - md = 20 - 16 - 6 = -2$

j = 2

$k_{1,2} = t_{1,2} - t_{1,1} - md = 37 - 32 - 6 = -1$
$k_{2,2} = 0$
$k_{3,2} = t_{3,2} - t_{3,3} - md = 94 - 87 - 6 = 1$
$k_{4,2} = t_{4,2} - t_{4,4} - md = 19 - 16 - 6 = -3$

j = 3

$k_{1,3} = t_{1,3} - t_{1,1} - md = 38 - 32 - 6 = 0$
$k_{2,3} = t_{2,3} - t_{2,2} - md = 67 - 60 - 6 = 1$
$k_{3,3} = 0$
$k_{4,3} = t_{4,3} - t_{4,4} - md = 21 - 16 - 6 = -1$

j = 4

$k_{1,4} = t_{1,4} - t_{1,1} - md = 41 - 32 - 6 = 3$
$k_{2,4} = t_{2,4} - t_{2,2} - md = 71 - 60 - 6 = 5$
$k_{3,4} = t_{3,4} - t_{3,3} - md = 98 - 87 - 6 = 5$
$k_{4,4} = 0$

$K^t =$

0	-1	0	3
3	0	1	5
2	1	0	5
-2	-3	-1	0

Kooperierende Zeitsynchronisationsverfahren

$C_j^r =$

32	.	.	.
69	60	.	.
95	.	87	.
20	.	.	16

32	37	.	.
.	60	.	.
.	94	87	.
.	19	.	16

32	.	38	.
.	60	67	.
.	.	87	.
.	.	21	16

32	.	.	41
.	60	.	71
.	.	87	98
.	.	.	16

$K^t =$

0	-1	0	3
3	0	1	5
2	1	0	5
-2	-3	-1	0

j = 1	j = 2	j = 3	j = 4	$\frac{\sum_{n}^{n} k_{i,j}}{n}$
0	-1	0	3	
3	0	1	5	
2	1	0	5	
-2	-3	-1	0	

= 3/3 = 1	= -3/3 = -1	= 0/3 = 0	= 13/3 = 4	Korrekturwerte
Uhr wird verlangsamt	Uhr wird beschleunigt	Uhr ohne Korrektur	Uhr wird verlangsamt	

Fehlertolerante Konvergenzalgorithmen:

- Fault-Tolerant Midpoint und
- Fault-Tolerant Average

– Fault-tolerant Midpoint:

Die *k* höchsten and *k* niedrigsten Werte bleiben unberücksichtigt.
Aus den Werten k+1 und n-k wird das arithmetische Mittel gebildet.

- Midpoint: (max_value + min_value) / 2
- das ist **nicht** der Median in der sortierten Werteliste!!!!

– Fault-tolerant Average:

Die *k* höchsten and *k* niedrigsten Werte bleiben unberücksichtigt. Aus den verbleibenden Werten wird das arithmetische Mittel gebildet.

Fault-Tolerant Clock Synchronization in Environments with High Message Delay Variation

Marcelo Moraes de Azevedo and Douglas M. Blough
Department of Electrical and Computer Engineering
University of California, Irvine – Irvine, CA 92717

September 1994

Technical Report ECE 94-09-01

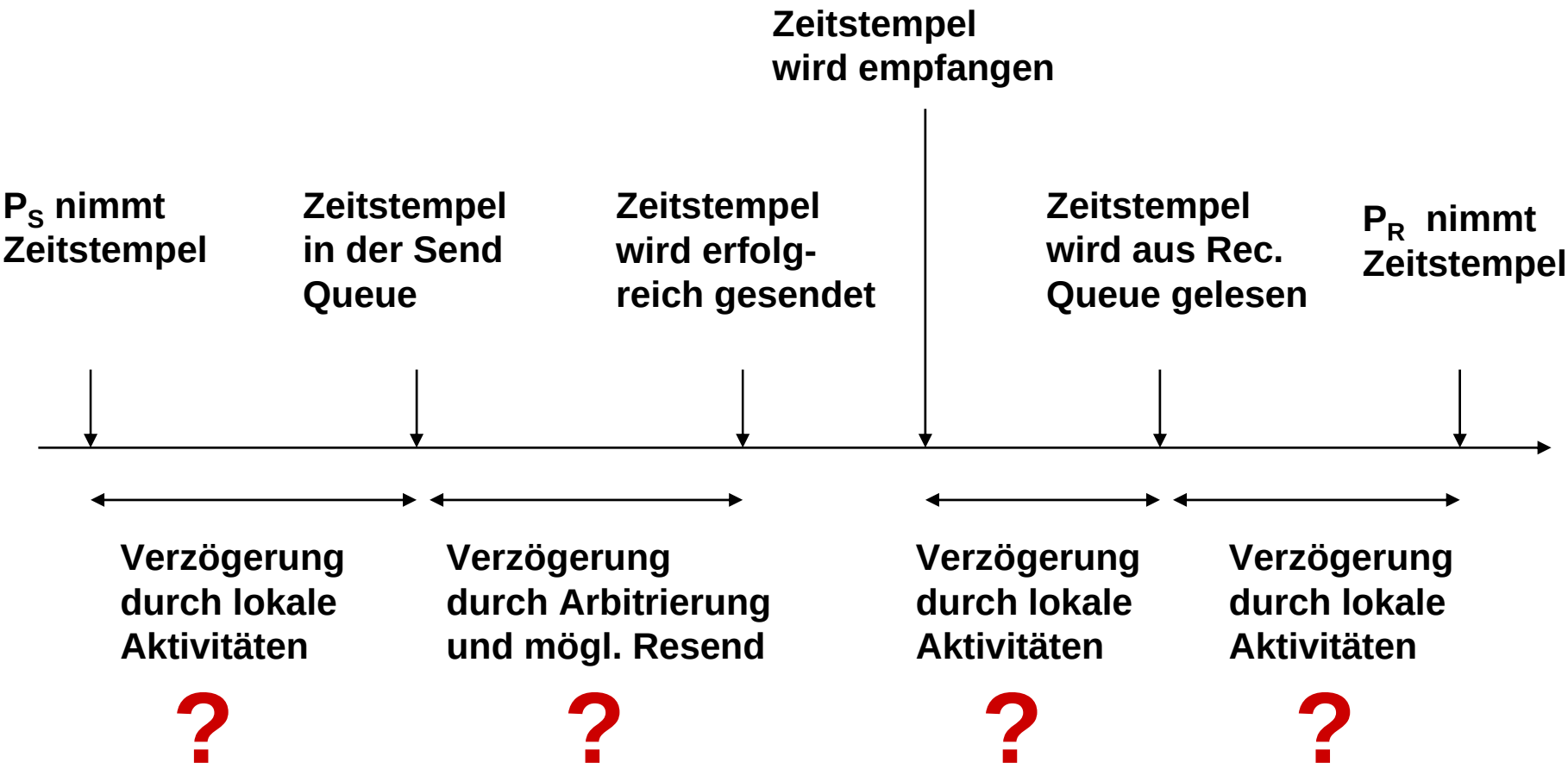
3.1 FTMA (Fault-Tolerant Midpoint Algorithm)

The Fault-Tolerant Midpoint Algorithm [3] relies on the hypothesis that at most k clocks are faulty at any resynchronization interval. At least $n = 3k+1$ nodes are required in the system to tolerate k Byzantine faults.

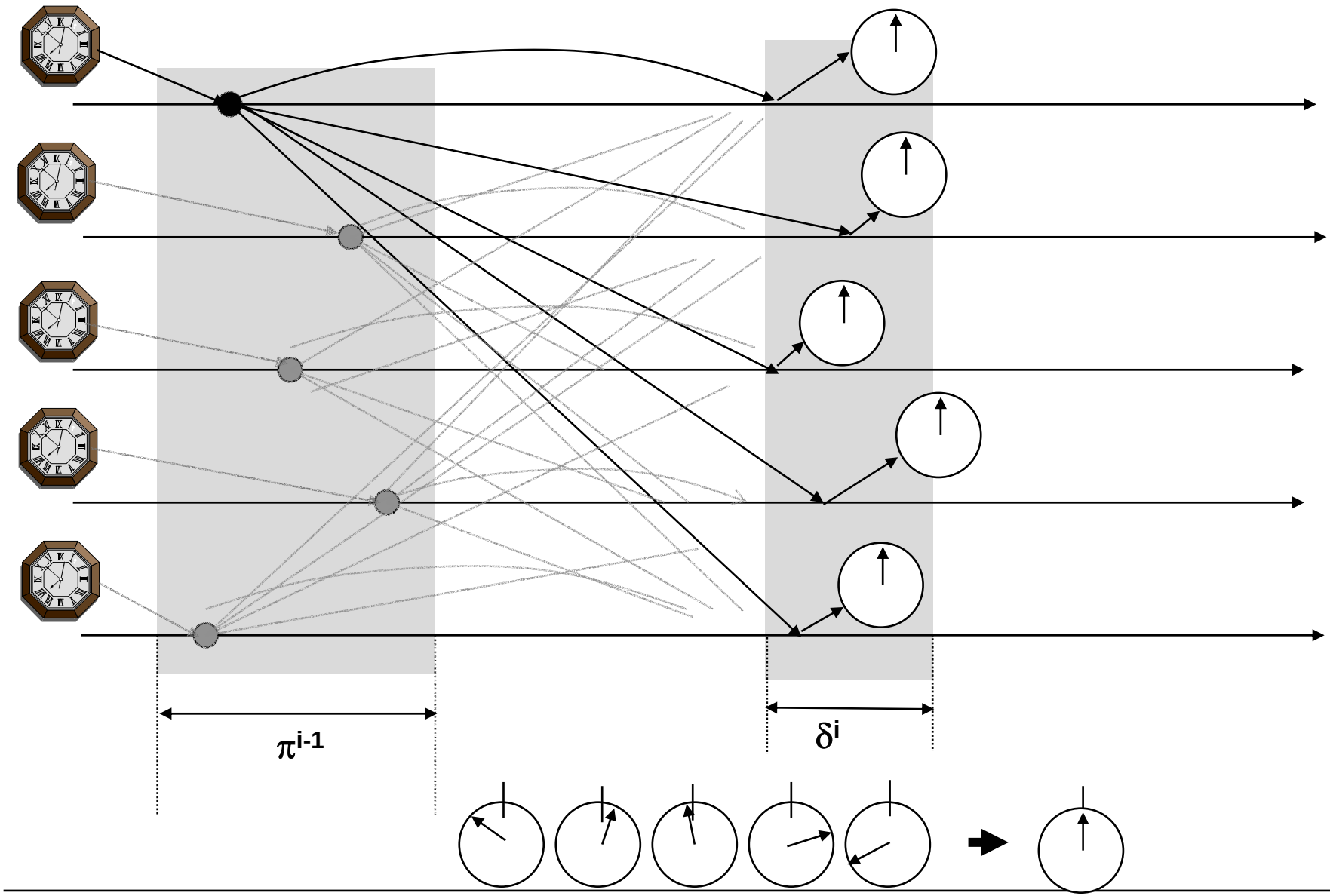
Assume that, at a given round r of the algorithm, a sorted clock deviation vector $\overline{X_j^r} = [x_1\ x_2\ \cdots\ x_i\ \cdots\ x_n]$, $x_i \leq x_{i+1}$, is available at node j (note that the elements in $\overline{X_j^r}$ are exactly the elements in $\overline{\Delta_j^r}$, except that in $\overline{X_j^r}$ they are sorted). To find the clock correction term, FTMA discards the k lowest and the k highest clock deviations and computes the arithmetic mean of x_{k+1} and x_{n-k}, i.e.

$$CORR_j^r = \frac{x_{k+1} + x_{n-k}}{2} \qquad (1)$$

Praktische Überlegungen:

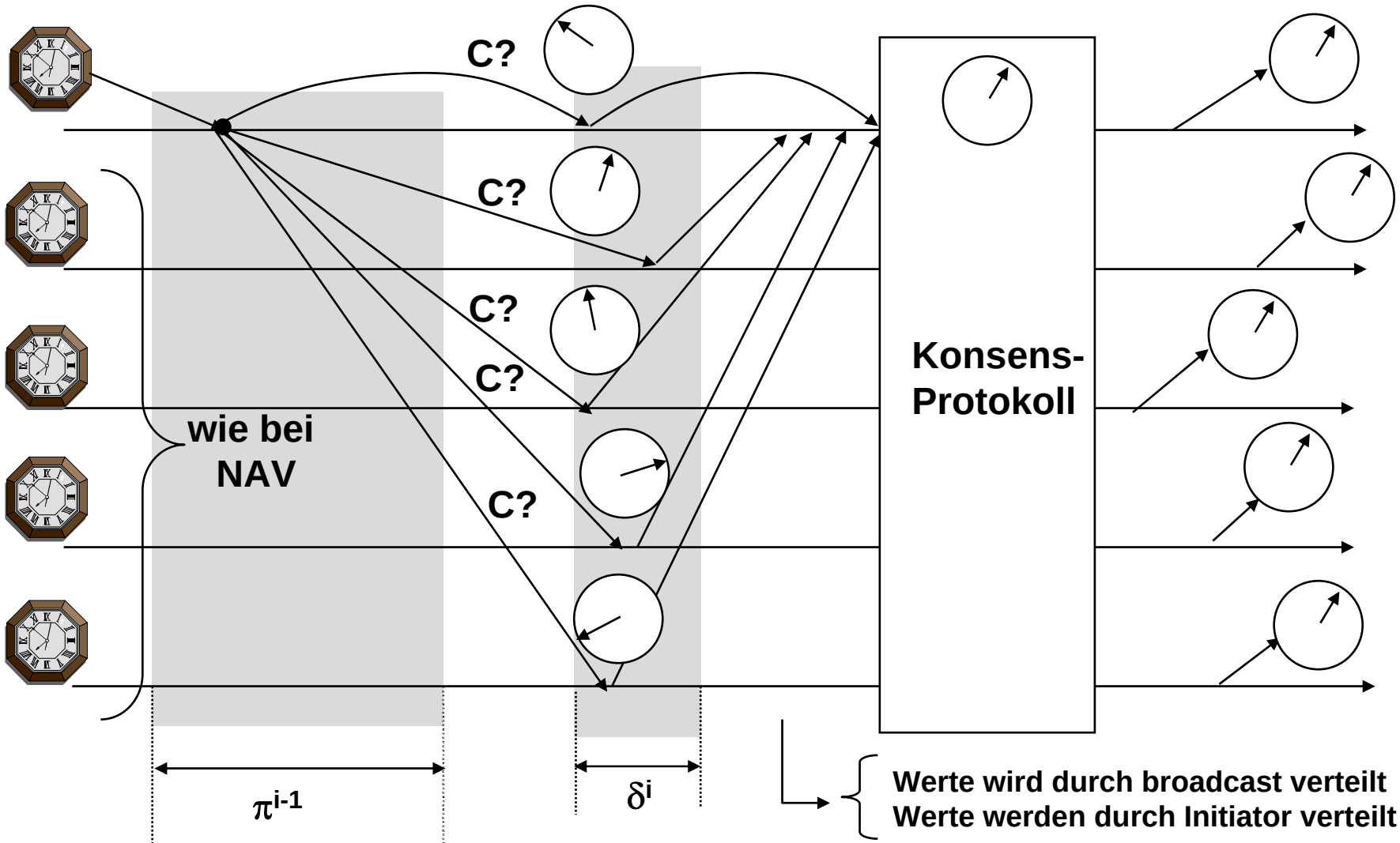


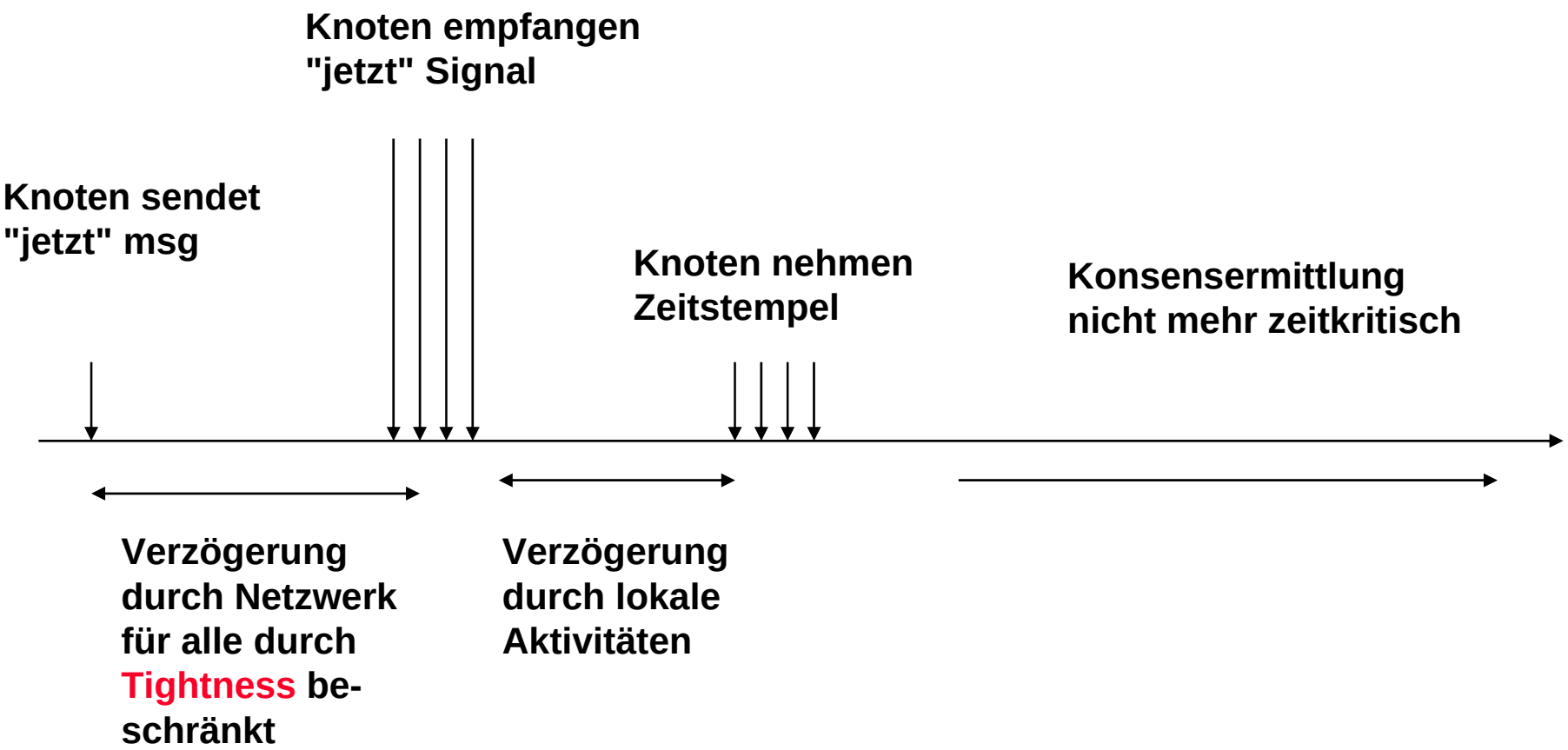
Nicht mittelndes Verfahren (non avaraging)



Hybrides Verfahren

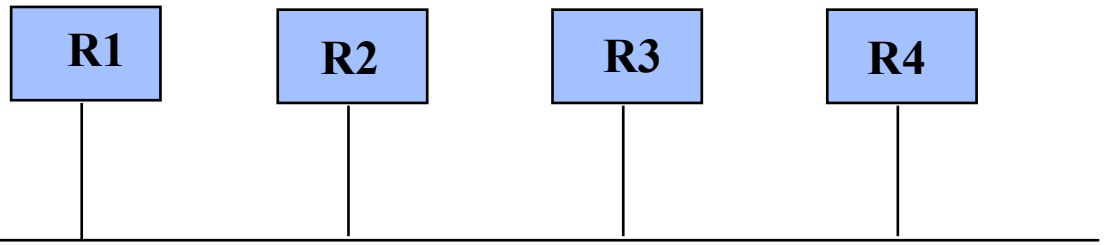
Uhr startet
Sync. Protokoll




Knoten empfangen
"jetzt" Signal
Knoten sendet
"jetzt" msg
Knoten nehmen
Zeitstempel
Konsensermittlung
nicht mehr zeitkritisch
Verzögerung
durch Netzwerk
für alle durch
Tightness be-
schränkt
Verzögerung
durch lokale
Aktivitäten

Kooperierende Zeitsynchronisationsverfahren

Beispiel:



md=6

$t_{1,1} = 32, 32$	$t_{1,2} = 37, 32$	$t_{1,3} = 38, 32$	$t_{1,4} = 41, 32$
$t_{2,1} = 69, 60$	$t_{2,2} = 60, 60$	$t_{2,3} = 67, 60$	$t_{2,4} = 66, 60$
$t_{3,1} = 95, 87$	$t_{3,2} = 94, 87$	$t_{3,3} = 87, 87$	$t_{3,4} = 98, 87$
$t_{4,1} = 20, 16$	$t_{4,2} = 19, 16$	$t_{4,3} = 21, 16$	$t_{4,4} = 16, 16$

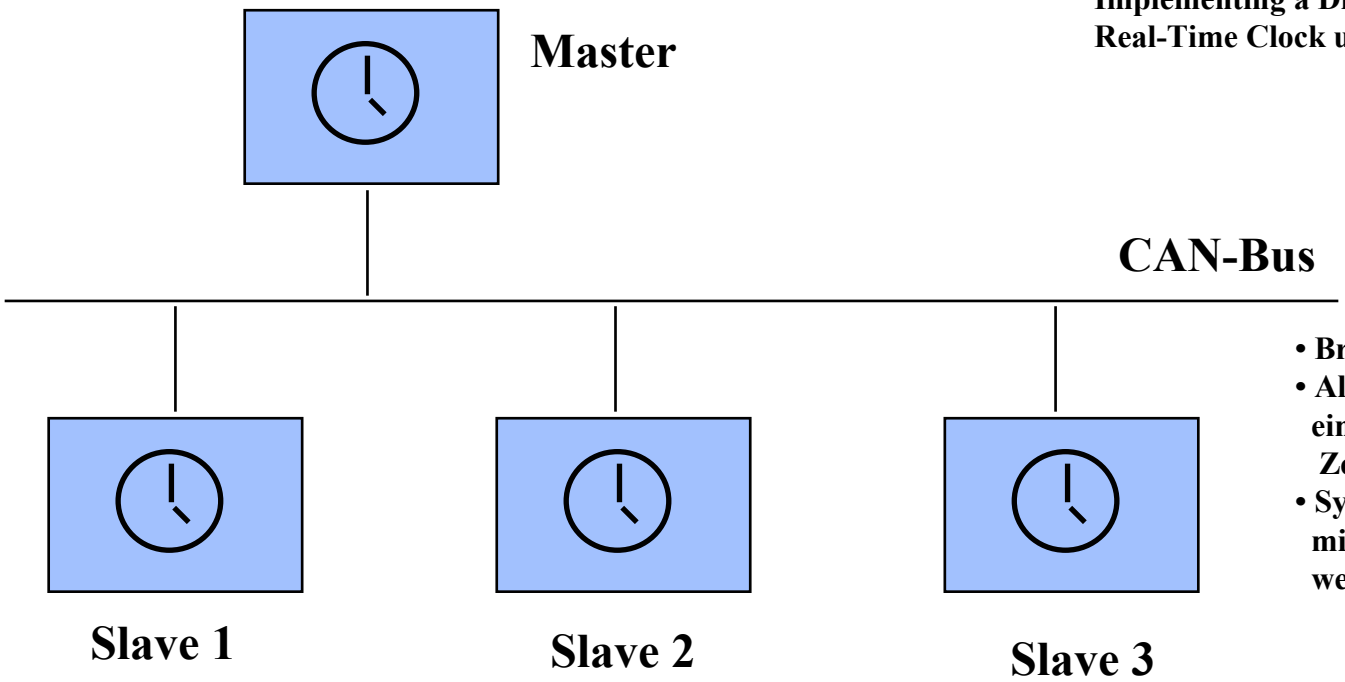
Zeitstempel enthält lokale Empfangszeit

$C_j^r =$

32 . . .	32 37 . .	32 . 38 .	32 . . 41
69 60 . .	. 60 . .	. 60 67 .	. 60 . 66
95 . 87 .	. 94 87 .	. . 87 .	. . 87 98
20 . . 16	. 19 . 16	. . 21 16	. . . 16
j=1	j=2	j=3	j=4

Ein Echtzeitsynchronisationsprotokoll für den CAN-Bus*

* M. Gergeleit, H. Streich
Implementing a Distributed High Resolution Real-Time Clock using the CAN-Bus



- Broadcast
- Alle Teilnehmer empfangen eine Nachricht zur selben Zeit
- Synch.- Nachrichten können mit hoher Priorität versehen werden

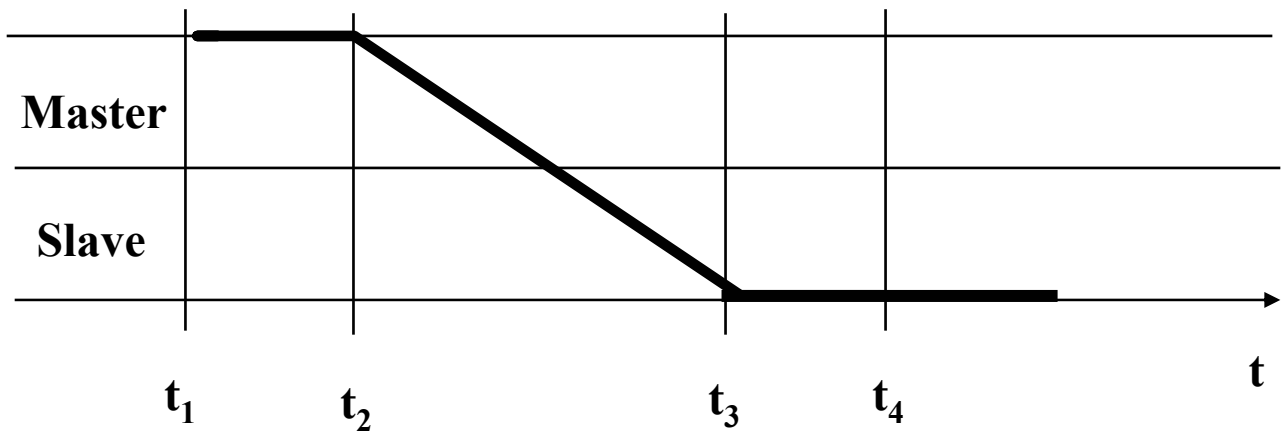
System:

- 1 Master, mehrere Slaves
- jeder Teilnehmer am Protokoll hat eine lokale Uhr
- lokale Uhren können gestellt werden
- alle Uhren gehen chronoskopisch

Ein Echtzeitsynchronisationsprotokoll für den CAN-Bus

Einfaches Protokoll

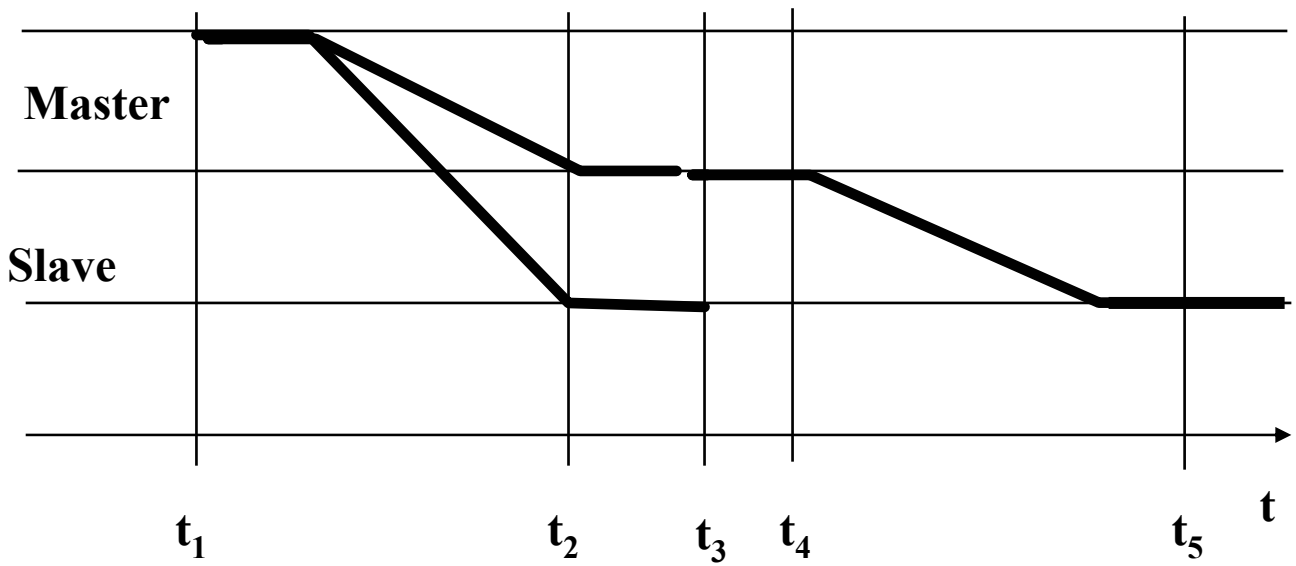
- Master sendet den Zeitstempel
- Slaves empfangen diese Nachricht
- Slaves gleichen ihre Zeit ab



Ein Echtzeitsynchronisationsprotokoll für den CAN-Bus

Verbessertes Protokoll

- Irgendein Rechner i sendet die Nachricht "jetzt" (t_1)
- Master und Slaves lesen ihre lokale Zeit beim Empfang der Nachricht (t_2)
- Master sendet seinen Zeitstempel (t_3, t_4)
- Slaves gleichen ihre Zeit danach ab (t_5)



Ein Echtzeitsynchronisationsprotokoll für den CAN-Bus

Optimiertes Protokoll

- **Master sendet die Nachricht "jetzt" (t_1)**
- **Die Nachricht enthält den Zeitstempel der letzten Sync.-Runde**
- **Master liest seine Uhrzeit, beim Empfangs-Interrupt (t_3)**
- **Slaves lesen ihre Uhrzeit, beim Empfangs-Interrupt (t_3)**
- **Slaves gleichen ihre Zeit nach dem letzten Zeitstempel des Masters ab (t_4)**