

# **Globale Zeit und Uhrensynchronisation in Netzen**

## Einige "Meilenstein"-Veröffentlichungen zur Uhrensynchronisation:

L. Lamport and P. M. Melliar-Smith. Synchronizing clocks in the presence of faults. Journal of the ACM, 32(1):5278, July 1985

F. Cristian, H. Aghili, and R. Strong: Clock synchronization in the presence of omission and performance failures, and processor joins. In Proc. of 16th International Symposium on Fault-Tolerant Computing Systems, July 1986

H. Kopetz and W. Ochsenreiter: Clock synchronization in distributed real-time computer systems. IEEE Transactions on Computers, C-36(8):933940, August 1987.

D. L. Mills. Internet time synchronization: The network time protocol. IEEE Transactions on Computers, 39(10):14821493, October 1991

P. Verissimo, A. Casimiro, and L. Rodrigues. Cesiumspray: a precise and accurate global time service for large scale systems. Journal of Real-Time Systems, 12:243294, 1997

### Guter Überblick:

Emmanuelle Anceaume and Isabelle Puaut: Performance Evaluation of Clock Synchronization Algorithms, Rapport de recherche N ° 3526, Octobre 1998  
ISSN 0249-6399

# Wozu dient Zeit ?

## **Orientierung:**

Einordnung in den kontinuierlichen „Zeitstrom“

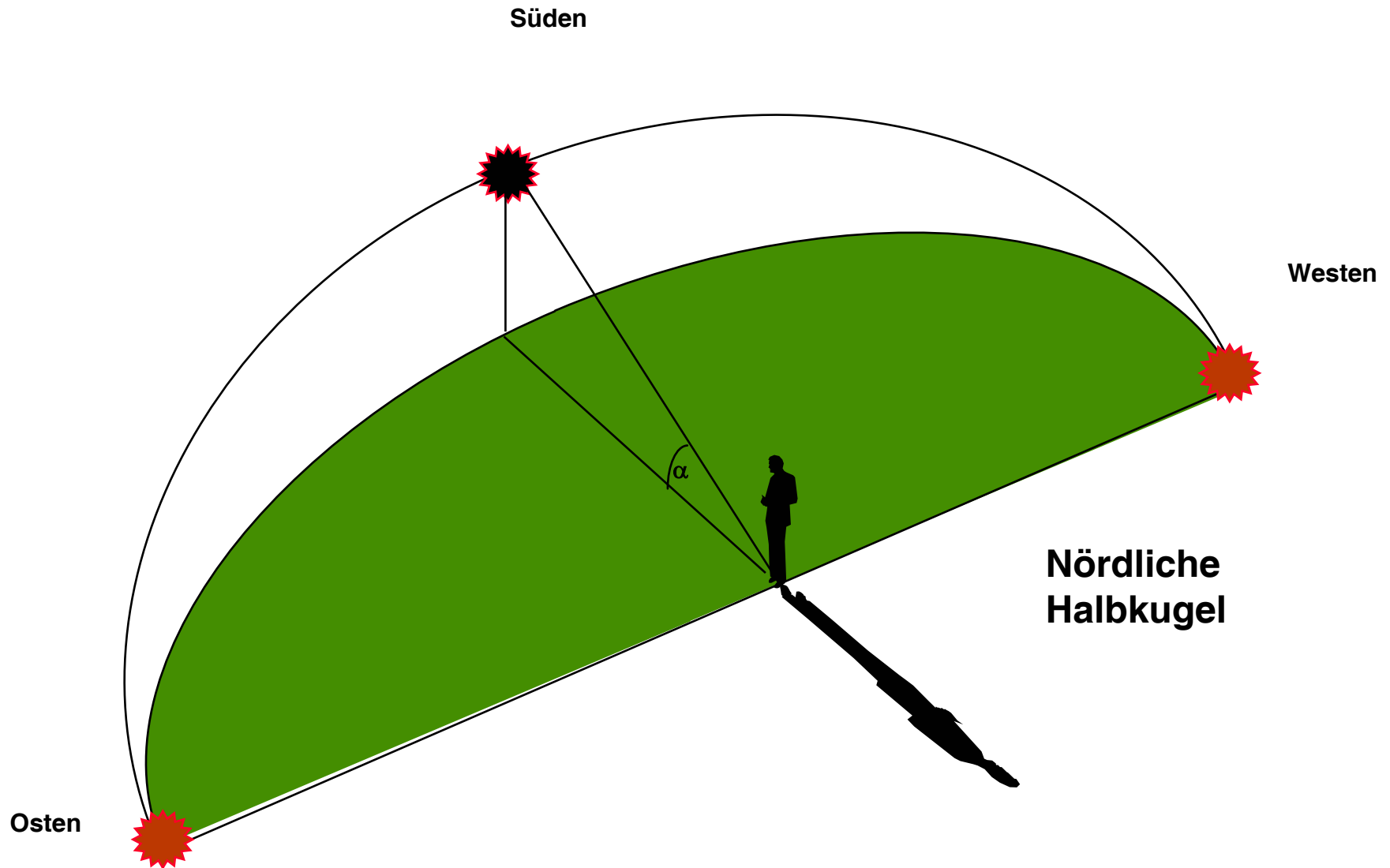
## **Regulierung**

Durchsetzung einer „Zeitdisziplin“

## **Koordinierung**

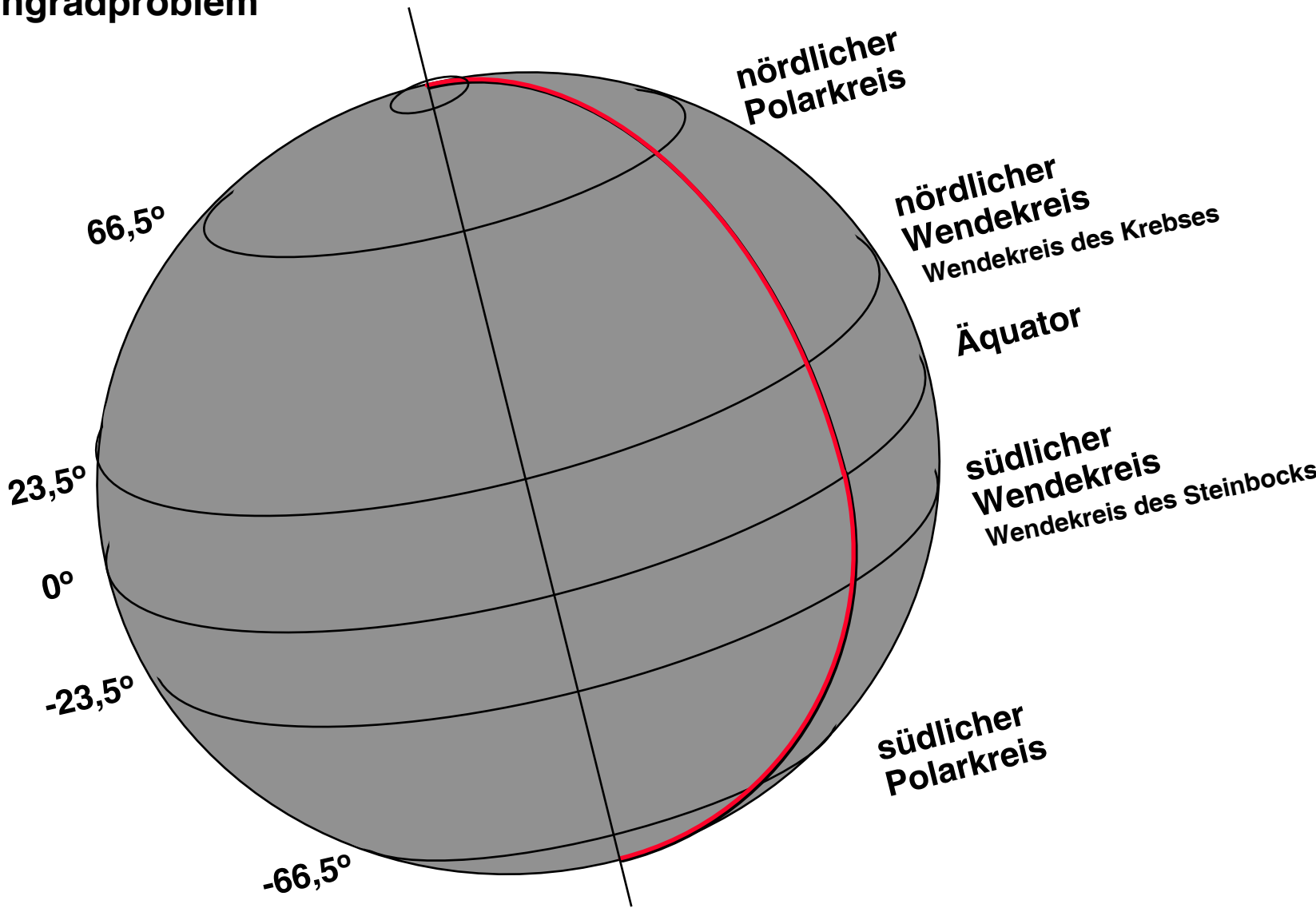
Abstimmung zur Durchführung gemeinsamer Aufgaben

# Wie wird die Zeit bestimmt?

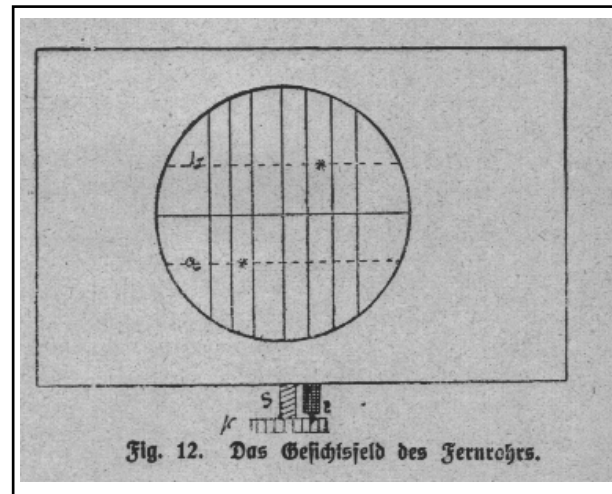
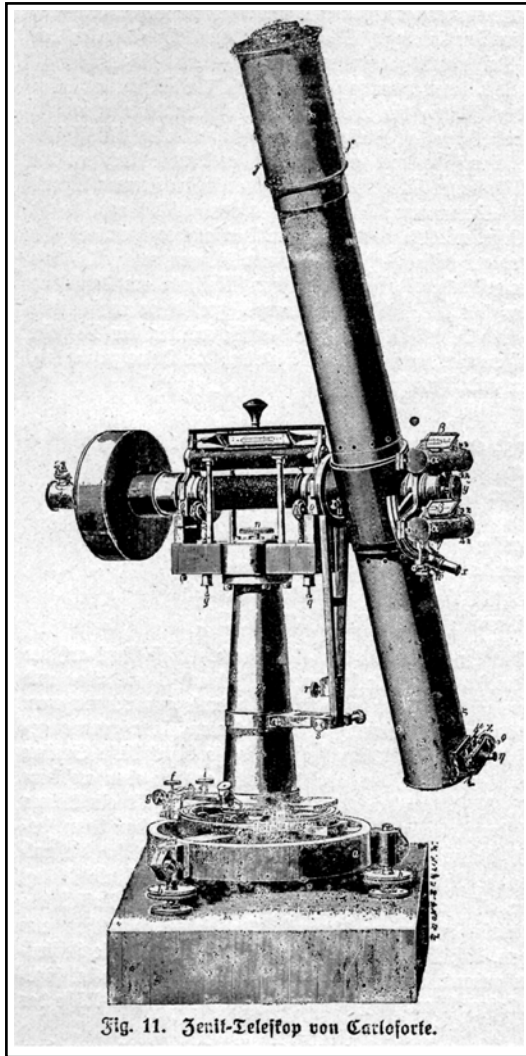


**Aus der Höhe (Elevation) der Sonne über dem Horizont läßt sich anhand eines Kalenders die geographische Breite ermitteln. Die höchste Elevation hat die Sonne um 12.00 Uhr Ortszeit.**

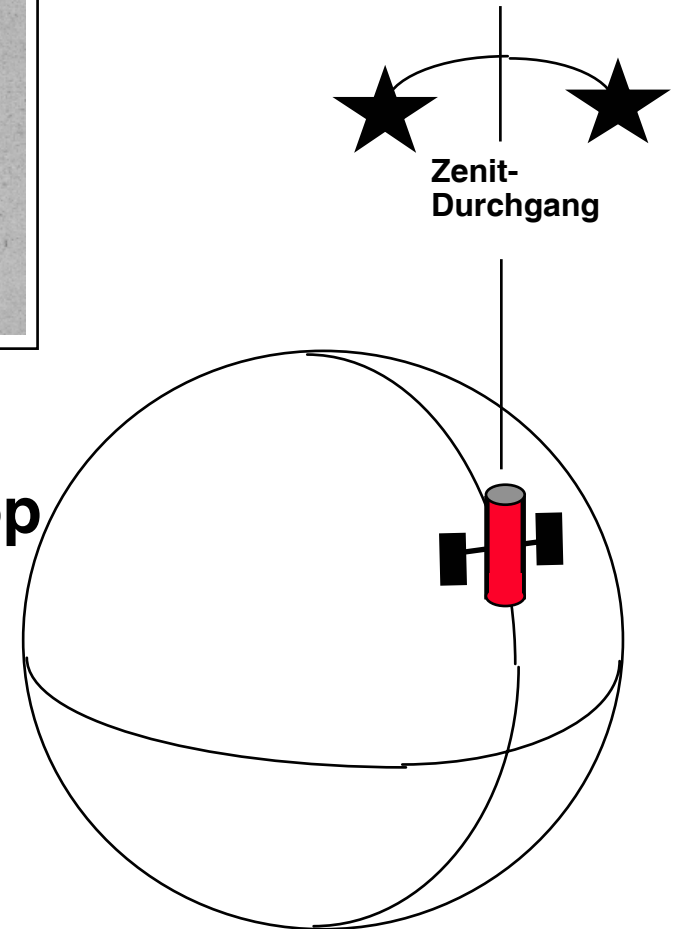
# Das Längengradproblem



# Bestimmung der Ortszeit durch Zenit-Teleskop



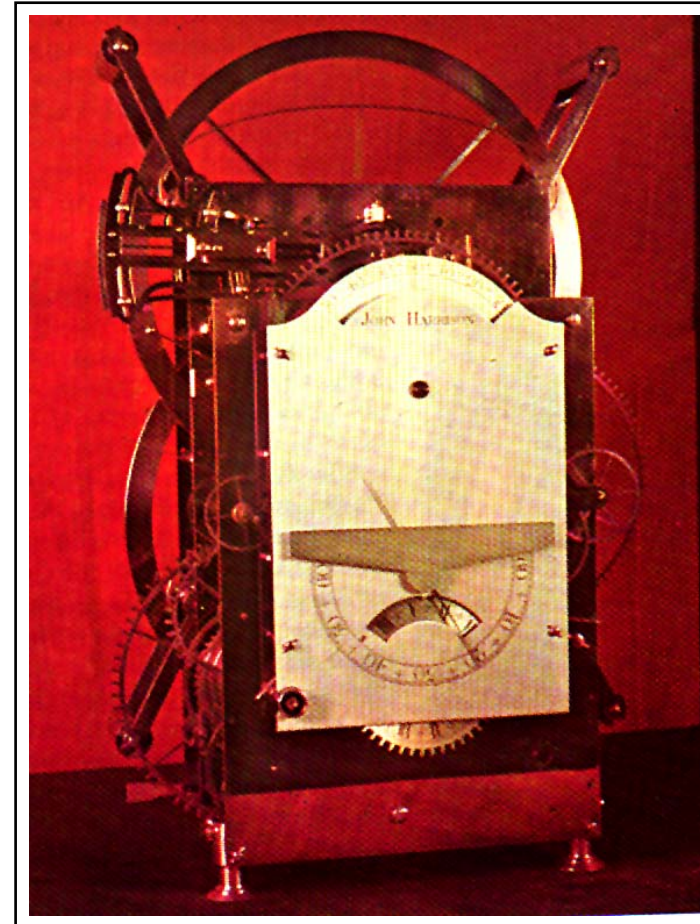
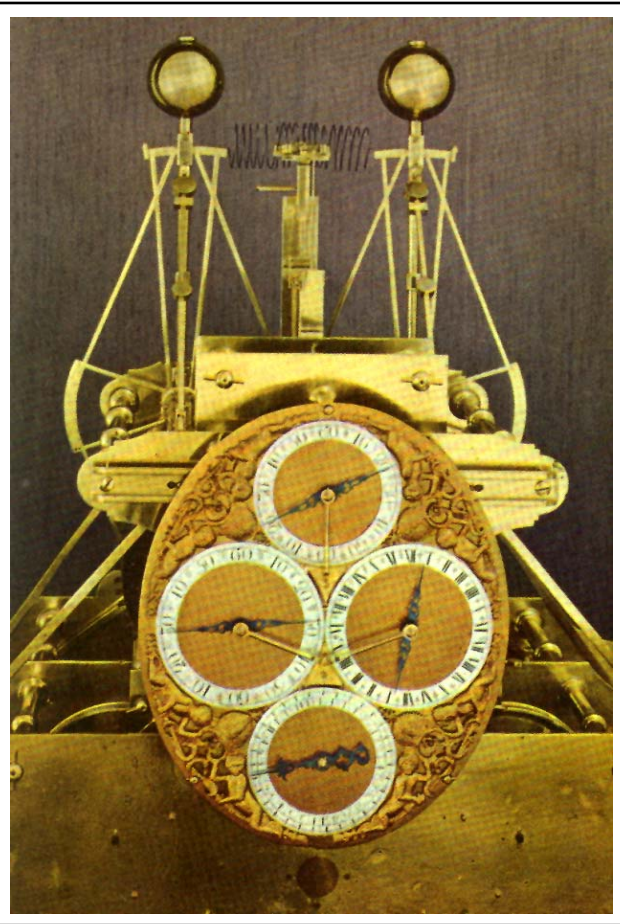
## Das Zenit-Teleskop



# Die Chronometer von Harrison

H1

H3



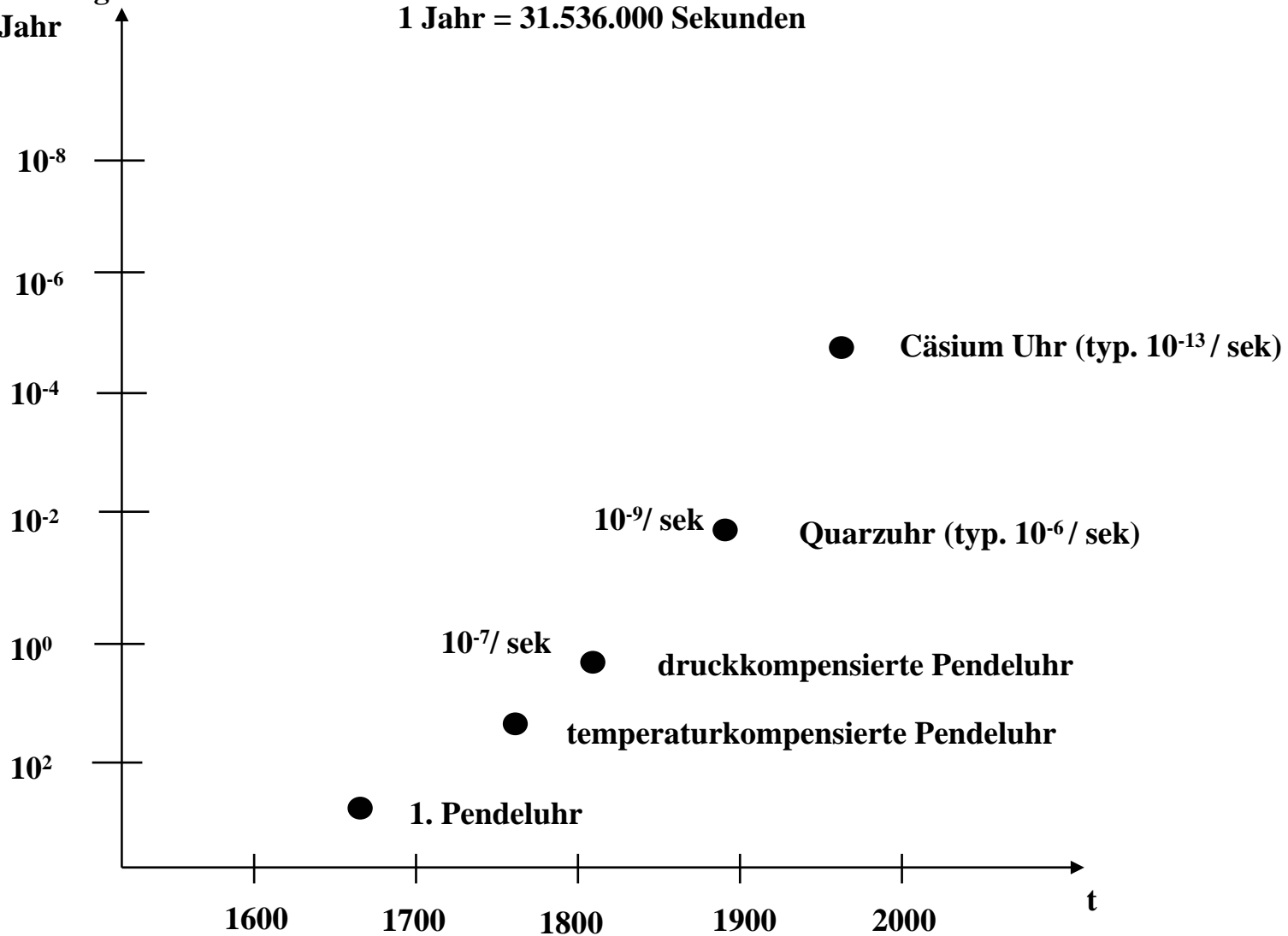
TIME-LIFE BOOKS: "Time", Time Inc., New York, 1970. Dt. Übersetzung: "Die Zeit", rororo-Sachbuch, erschienen im Rowohlt Taschenbuch Verlag GmbH, Reinbeck bei Hamburg, September 1970

"The New Caxton Encyclopedia, Vol. 5", The Caxton Publishing Company Ltd. London, New York, Melbourne, Toronto Wellington, 1969. First published under the title: "Purnells New English Encyclopedia", Purnell and Sons Ltd. 1969

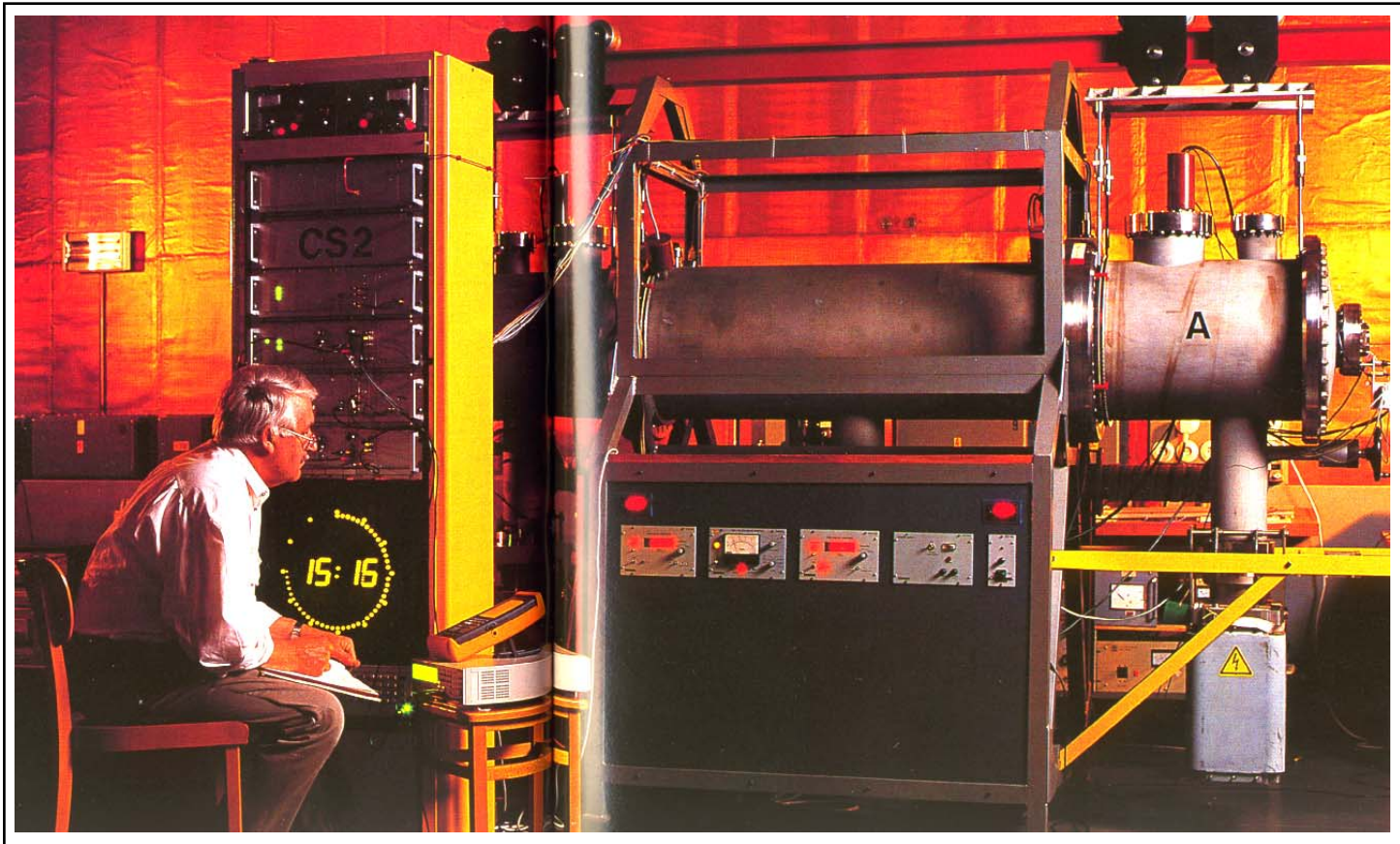
# Fortschritte in der Zeitmessung

1 Jahr = 31.536.000 Sekunden

Abweichung  
Sek/Jahr







## 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

---

<b>Astronomische Zeit (AT)</b>	basiert auf der gleichförmigen Bewegung von Himmelskörpern
<b>Physikalische Zeit (PT)</b>	basiert auf periodischen physikalischen Prozessen (Schwingungen)
<b>Sonnendurchlauf:</b>	Höchster Punkt der Sonne während eines Tages
<b>Sonnentag:</b>	Intervall zwischen zwei aufeinanderfolgenden Sonnendurchläufen
<b>Sonnensekunde:</b>	Der 1 / 86400 Teil eines Sonnentages
<b>Mittlere Sonnensekunde:</b>	Sonnensekunde gemittelt über eine große Anzahl von Sonnentagen
<b>Zeitzone:</b>	Erster Zeitstandard (1820) basiert auf mittlerer Sonnensekunde
	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.
<b>UT (AT, 1833)</b>	Universal Time (UT) Mittlere Sonnenzeit, gemessen am Greenwich 0-Meridian (GMT). Basiert auf der mittleren Länge eines Sonntags, d.h. auf der Erdrotation
<b>ET (AT, 1955)</b>	Ephemeridenzeit (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 aufeinanderfolgenden Durchläufen der Sonne durch den Himmelsäquator in derselben Richtung.)
<b>UT0 (AT, 1960)</b>	Zeit, basierend auf den lokalen Beobachtungen verschiedener, über die Erde verteilter Observatorien.
<b>UT1 (AT, 1960)</b>	Zeit, basierend auf der Koordination der verschiedenen UT0-Zeiten (Mittlung).
<b>UT2 (AT, 1960)</b>	Nochmals, auf empirischer Basis korrigierte UT1-Zeit.
<b>TAI (PT, 1961)</b>	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).
<b>UTC (PT, 1972)</b>	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, standardisierte Zeitreferenz: *Universal Coordinated Time (UTC)***

**verteilt durch**

- **landgestützte Radiosender (0,1-10 msec)**
- **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  $\pm 5\text{ns}$  synchronisiert.
  
- Standortbestimmung basiert auf Abstandsmessung durch Laufzeitberechnung:  
 $R = (T_o - T_s) \cdot c$  (1,5m bei Abweichung von 5ns, 300m bei 1  $\mu\text{s}$ )
  
- 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 zueinander in 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 Zeitskala  $t_{e1} \leq t_{e2}$  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.

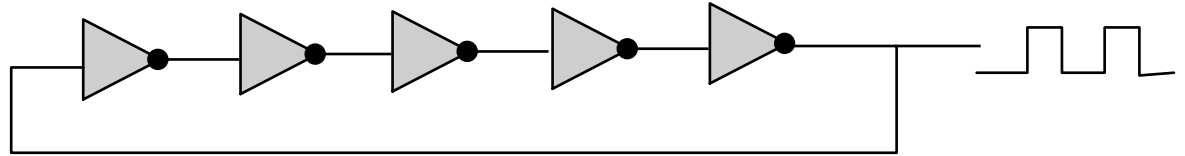
# Physikalische Uhren:

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

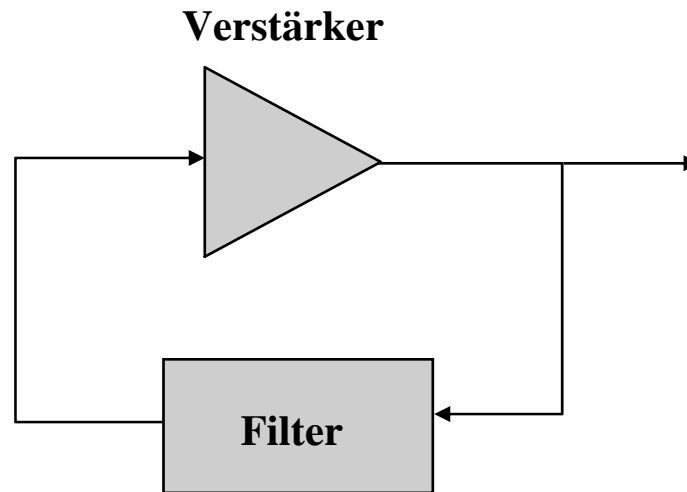
## Charakterisierung physikalischer Uhren:

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

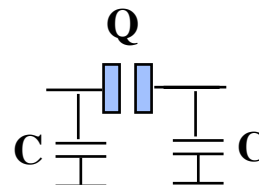
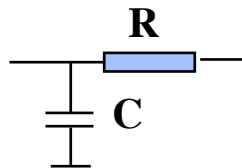
## Einfacher Oszillator



## Oszillator



Alternativen, z.B.:





Sei  $C(t)$  die Funktion, welche einen Zeitpunkt  $t$  der Referenzzeit 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,

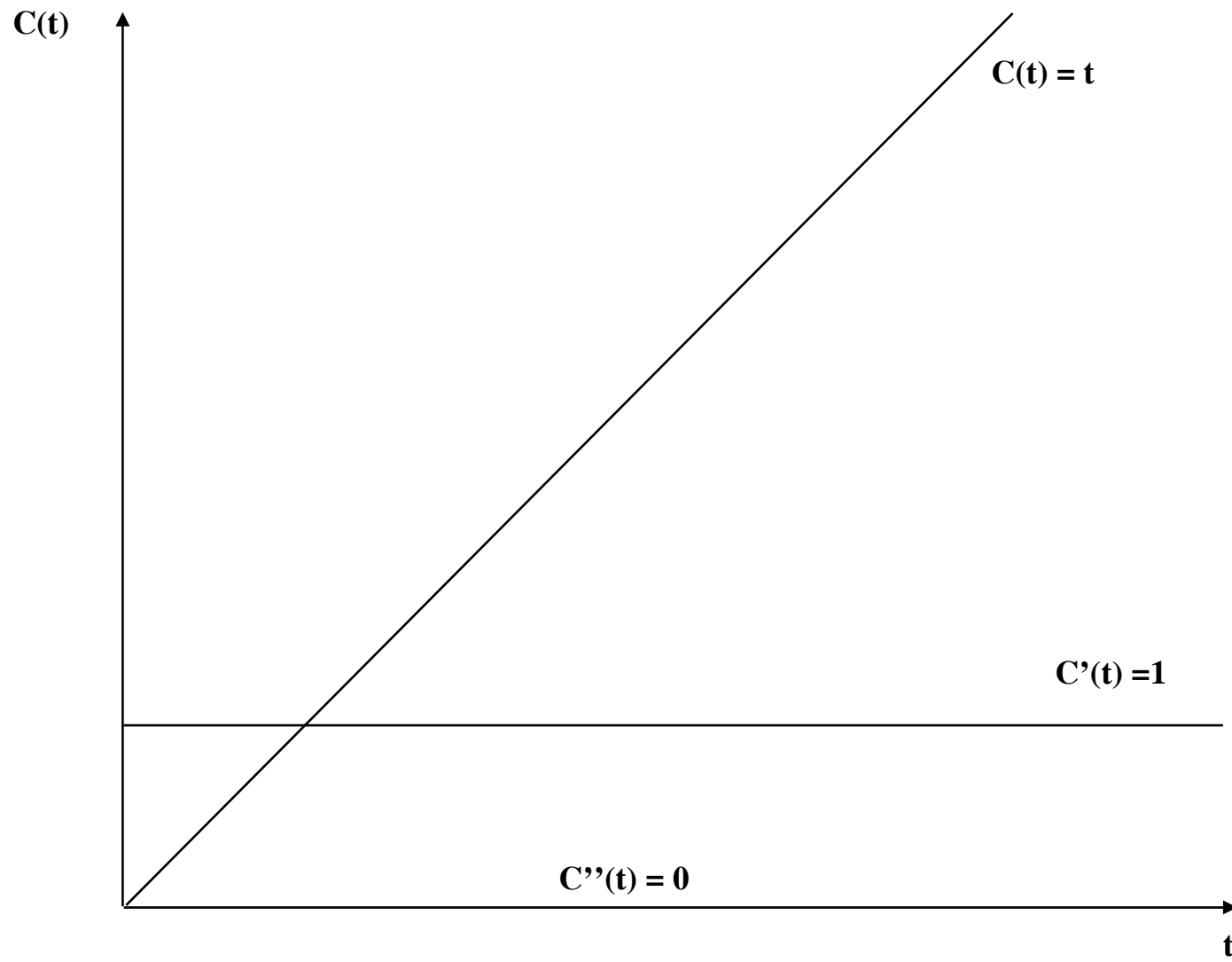
$dC/dt = 1$ : Uhr geht genau,

$dC/dt < 1$ : Uhr geht zu langsam

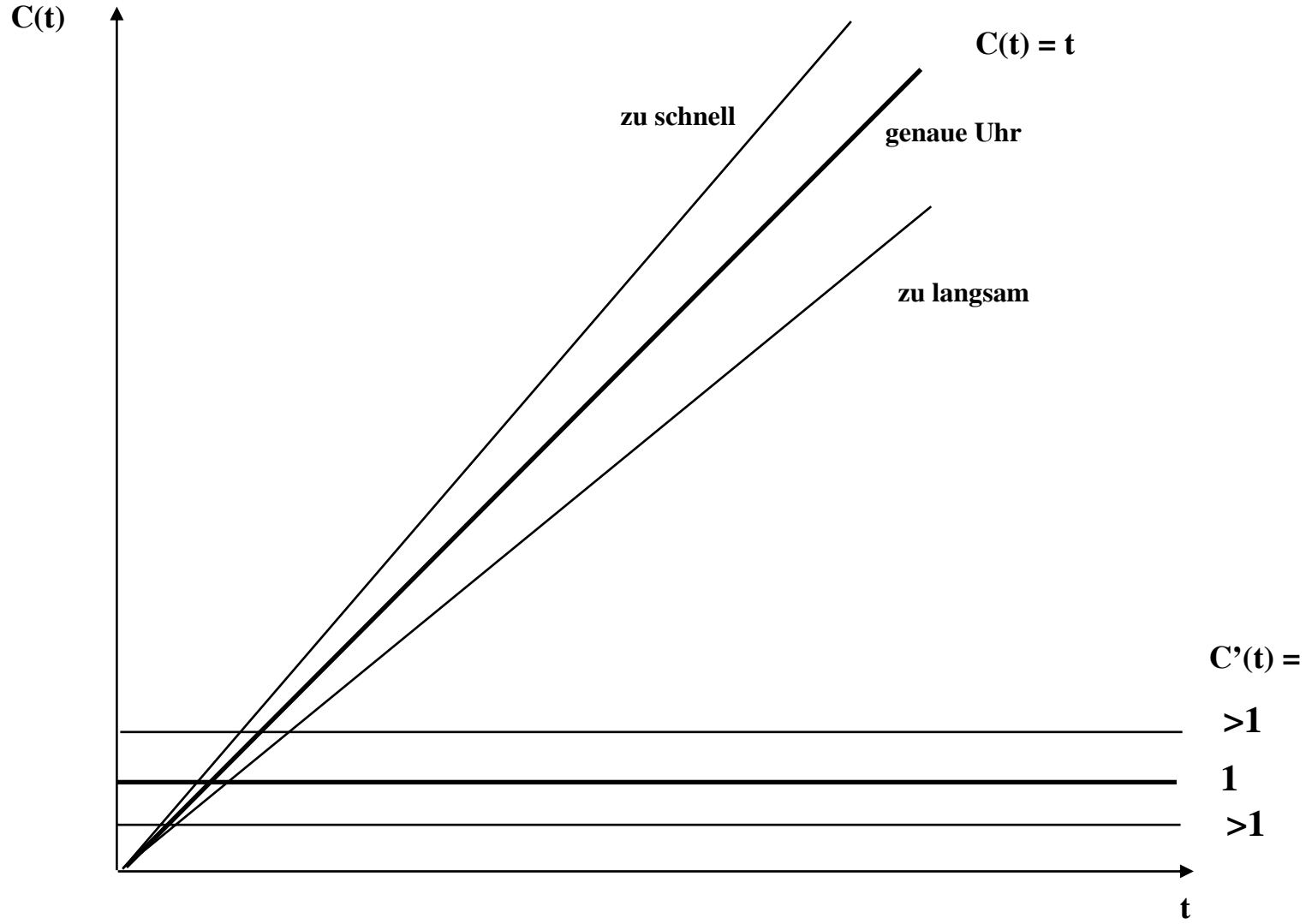
**Drift(rate)  $r$  : Abweichung der Uhr / Zeit**

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

# Die ideale Uhr



# Uhr mit konstanter Drift

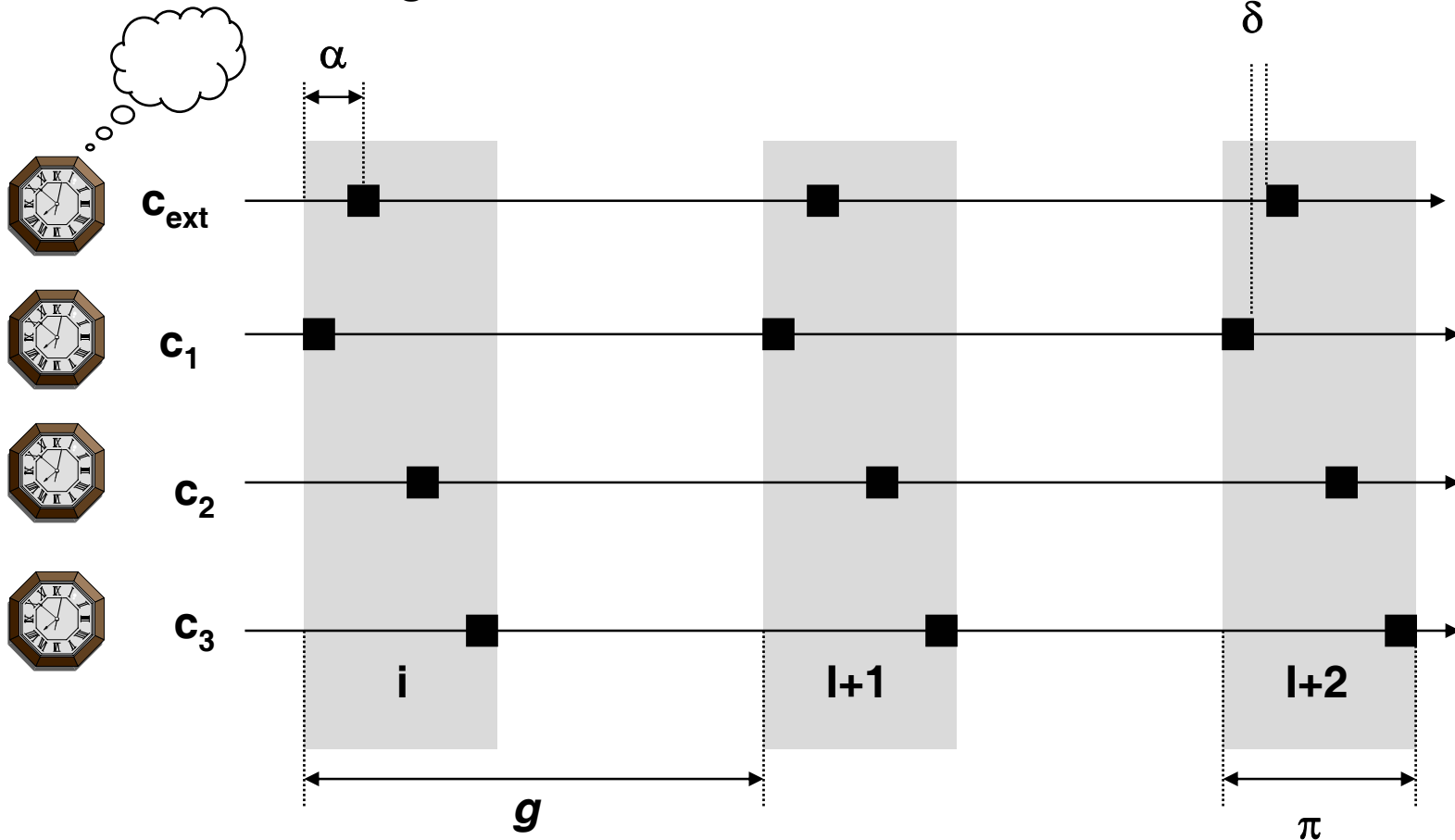


# Terminologie

---

<b>Ganggenauigkeit, (stability)</b>	<b>:</b>	<b>Abweichung von idealer Frequenz</b>
<b>Zeitgenauigkeit <math>\alpha</math> (accuracy)</b>	<b>:</b>	<b>Abweichung von einem Zeitnormal</b>
<b>Granularität <math>g</math> (granularity)</b>	<b>:</b>	<b>Kleinste meßbare Zeitdifferenz</b>
<b>Gleichlauf <math>\pi</math> (precision)</b>	<b>:</b>	<b>Abstand zweier Uhren voneinander</b>
<b>Konvergenz</b>	<b>:</b>	<b>Abstand zweier Uhren voneinander sofort nach der Synchronisation</b>
<b>Offset <math>\delta</math></b>	<b>:</b>	<b>Zeitdifferenz zwischen zwei Uhren</b>
<b>Skew</b>	<b>:</b>	<b>Frequenzdifferenz zwischen zwei Oszillatoren</b>
<b>Drift <math>\rho</math></b>	<b>:</b>	<b>Langsames Entfernen von einer Referenzzeit</b>

# Externes Zeitsignal



- $\alpha$  : accuracy (Zeitgenauigkeit)
- $\pi$  : precision (Gleichlauf)
- $g$  : granularity (Granularität)
- $\delta$  : offset (Differenz zweier Uhren)

# Uhrensynchronisation

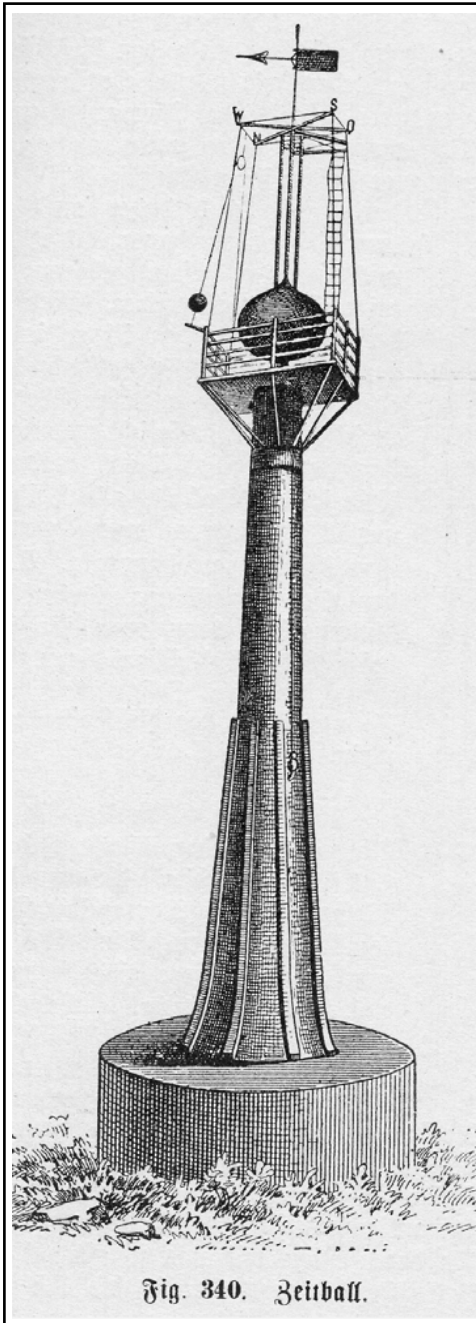
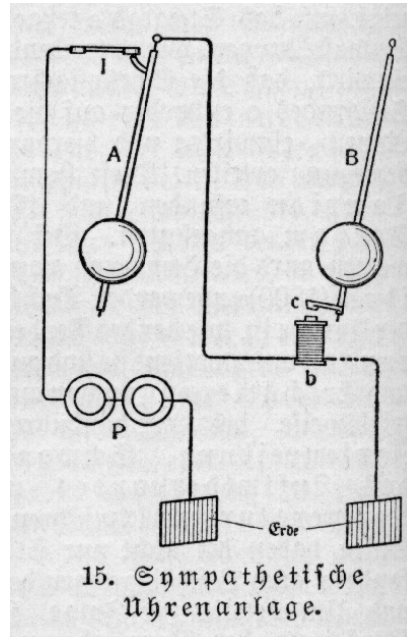
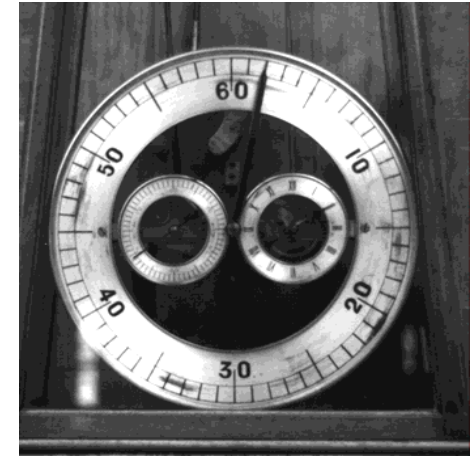


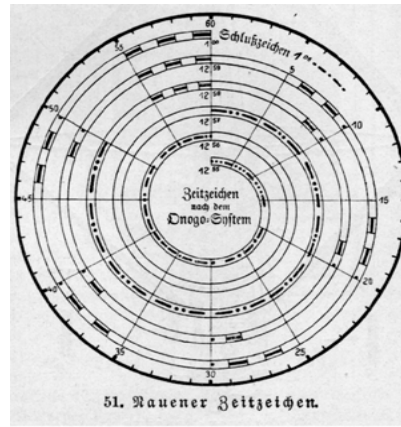
Fig. 340. Zeitball.



15. Sympathetische Uhrenanlage.



Direkte elektromagnetische Synchronisation



51. Rauener Zeitzeichen.

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  $\rho$ :

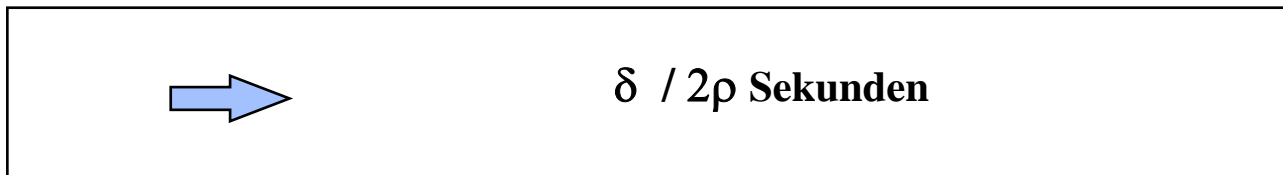
$$1-\rho \leq dC/dt \leq 1+\rho$$

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

Abstand zweier synchronisierter Uhren nach  $\Delta t$ :

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

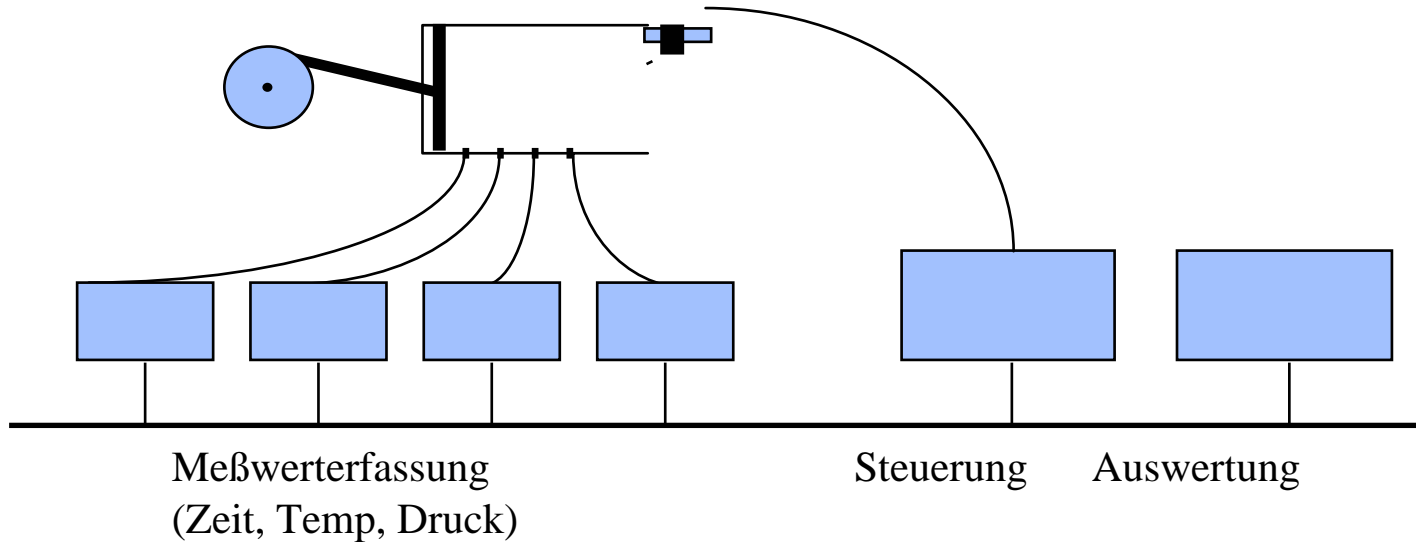
Resynchronisation zweier Uhren, die nicht weiter als  $\delta$  abweichen sollen, nach





## Drift physikalischer Rechneruhren

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



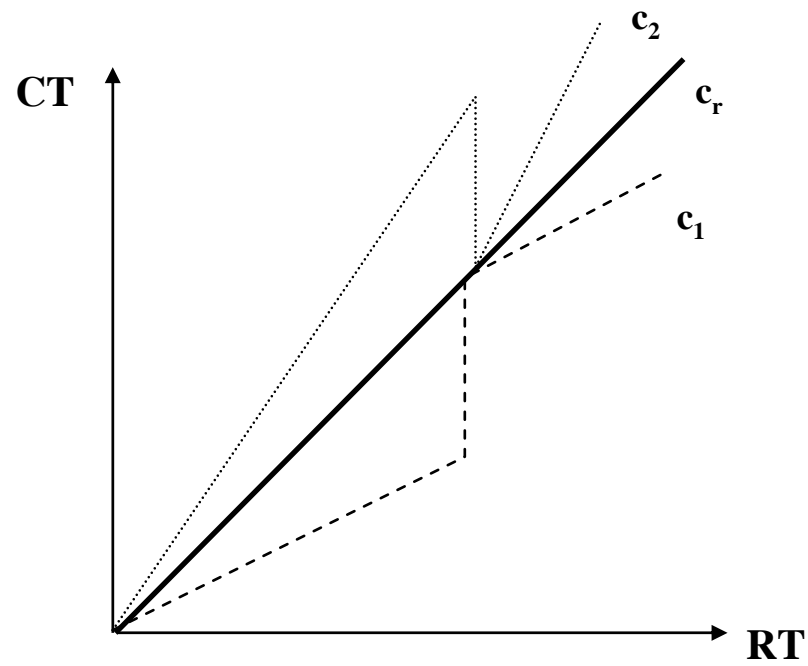
Uhr mit einer Frequenz von 10 Mhz; Periode = 100ns

Nach 10 Sek Versuchsdauer ist der worst-case Abstand zwischen zwei Uhren:

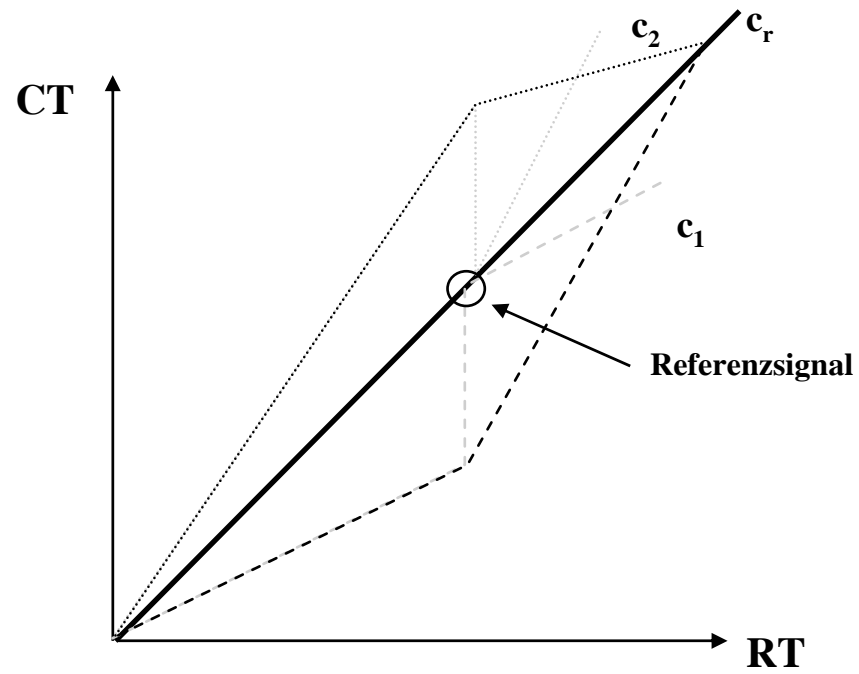
$$10 * 2 * 10^{-6} = 2 * 10^{-5} = 20 \mu\text{sek} = 20.000 \text{ nsek}$$

➔ 200-fache Auflösungsvermögen der Uhr!!

# Wie Synchronisation ?



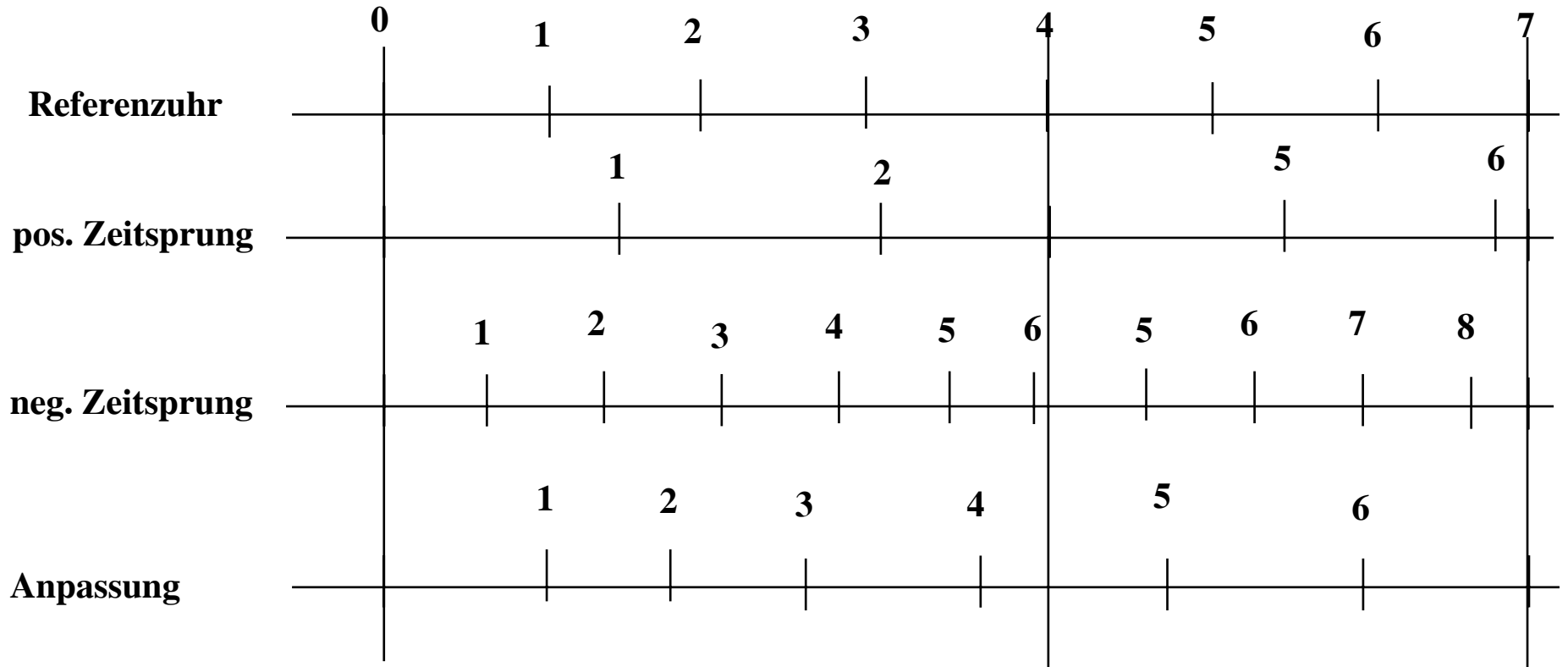
# Wie Synchronisation ?



# Grundlagen der Zeitsynchronisation

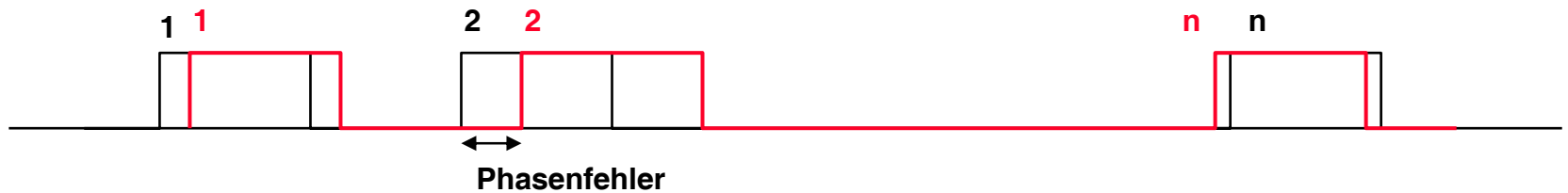
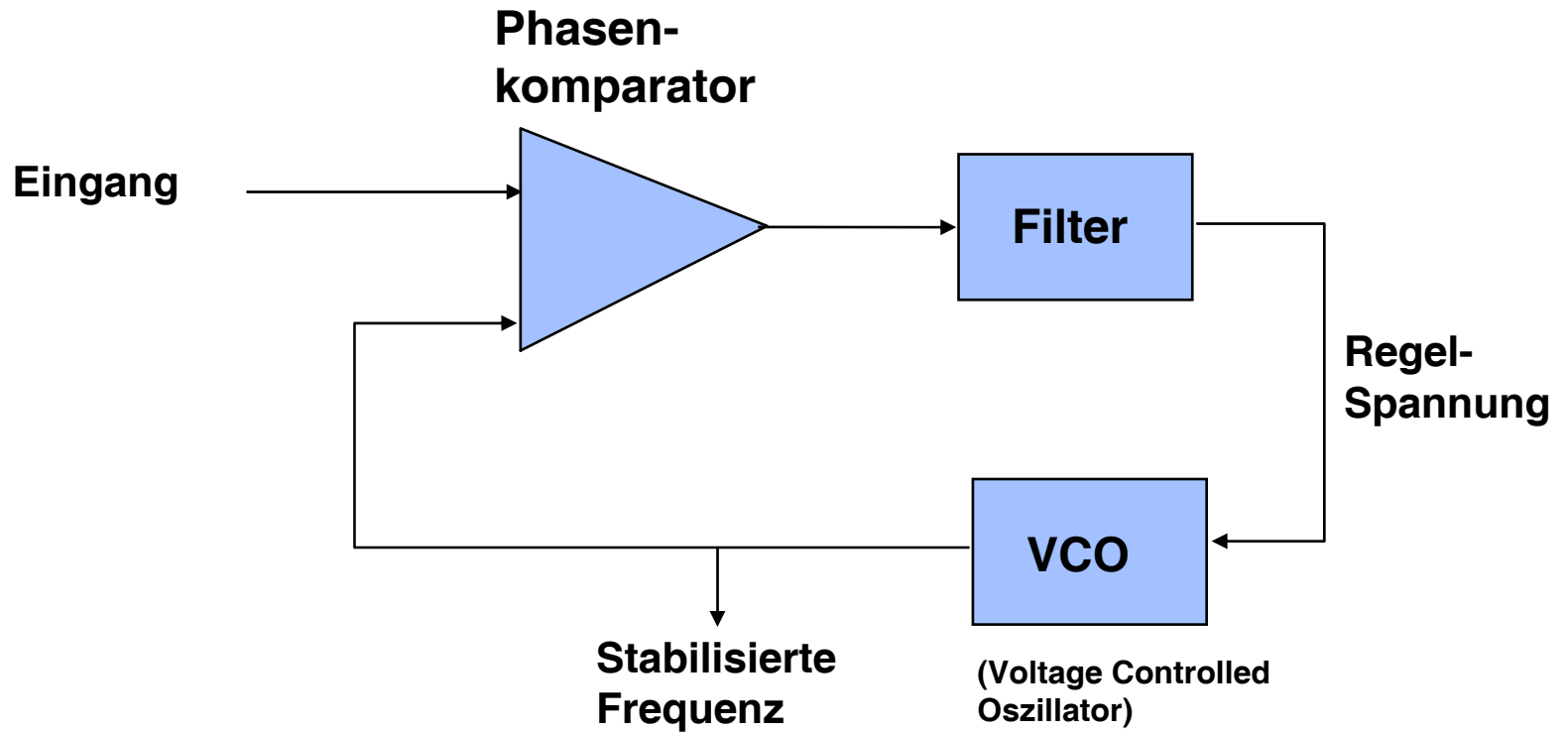
Korrektur des Zählerstandes

Korrektur der Drift



# PLL: Phase Locked Loop

## Phasenverriegelte Kontrollschleife



# **Uhrensynchronisation über Nachrichten in allgemeinen Kommunikationsnetzen**

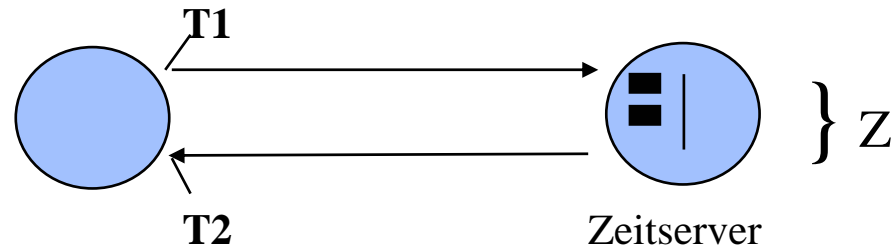
- **Zeitserver**
- **Dezentrale und kooperative Verfahren**

# 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)

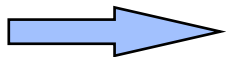


(a) Anfordern der Zeit beim Zeitserver

(b) Korrektur des erhaltenen Wertes um Korrekturfaktor **K**

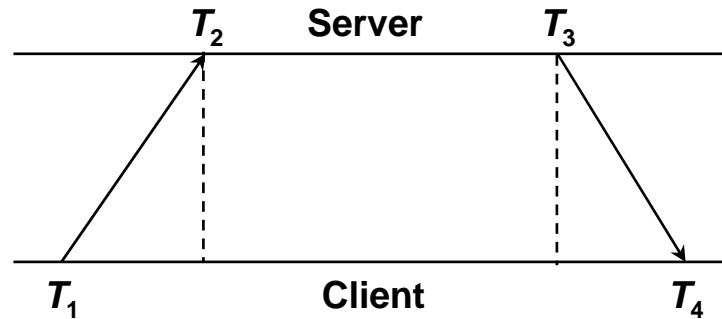
Probleme:

- 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 Zeitserver



$$\mathbf{K} = (\mathbf{T2} - \mathbf{T1} - \mathbf{Z}) / 2$$

## Berechnung des Offsets $o$ und der Verzögerung $d$

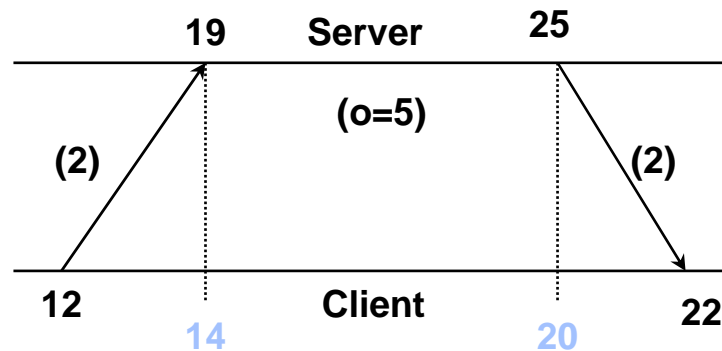


$$o = [(T_2 - T_1) + (T_3 - T_4)] / 2$$

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



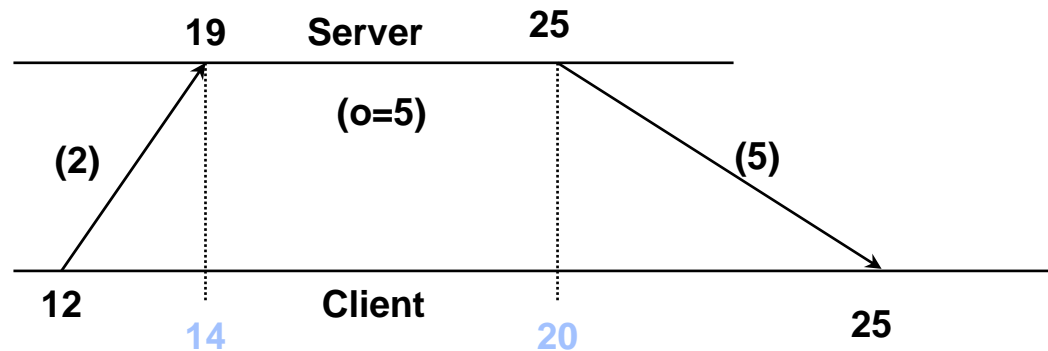
## Berechnung des Offsets $o$ und der Verzögerung $d$



$$o = [(19 - 12) + (25 - 22)] / 2$$
$$o = 5$$

$$d = (22 - 12) - (25 - 19)$$
$$d = 4$$

## Berechnung des Offsets $o$ und der Verzögerung $d$



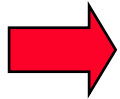
$$o = [(19 - 12) + (25 - 25)] / 2$$

$$o = 7$$

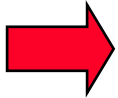
$$d = (25 - 12) - (25 - 19)$$

$$d = 7$$

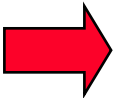
## Parameter bei der Berechnung der Zeit:



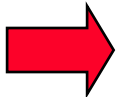
**Nachrichtenlaufzeiten auf dem Kommunikationsnetzwerk  
(steadyness + tightness)**



**Zeit bis Nachricht verschickt wird  
Verzögerungen in der Reply Queue**



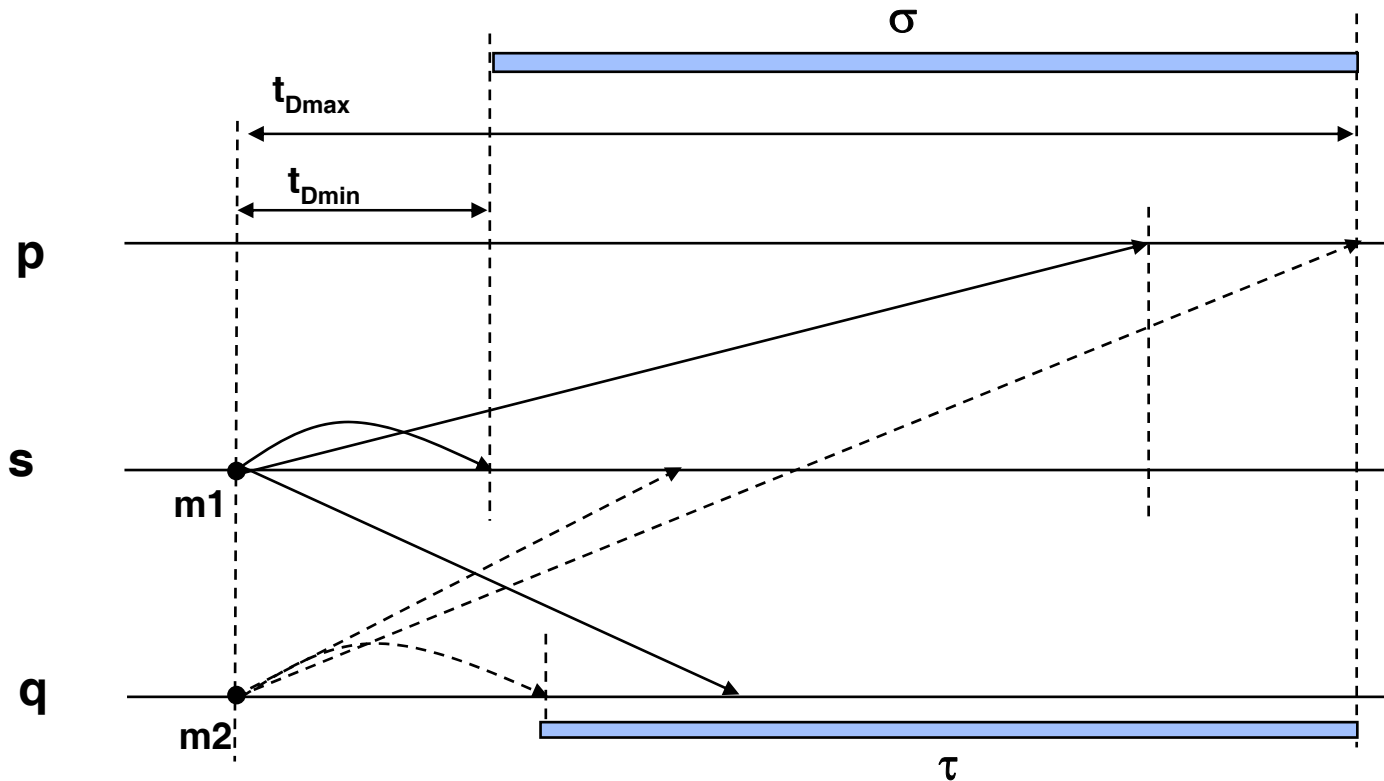
**Lokale Auslesefehler der Uhren  
Verzögerungen in der Request Queue  
Zeitabweichungen  
Zugriffsverzögerungen**



**Physische Parameter der Serveruhr:  
Drift, Offset**

# Zur Erinnerung:

Steadyness  $\sigma$  : Unterschied der Laufzeiten **verschiedener** Nachrichten (zu versch. Knoten).  
Tightness  $\tau$  : 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 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): 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

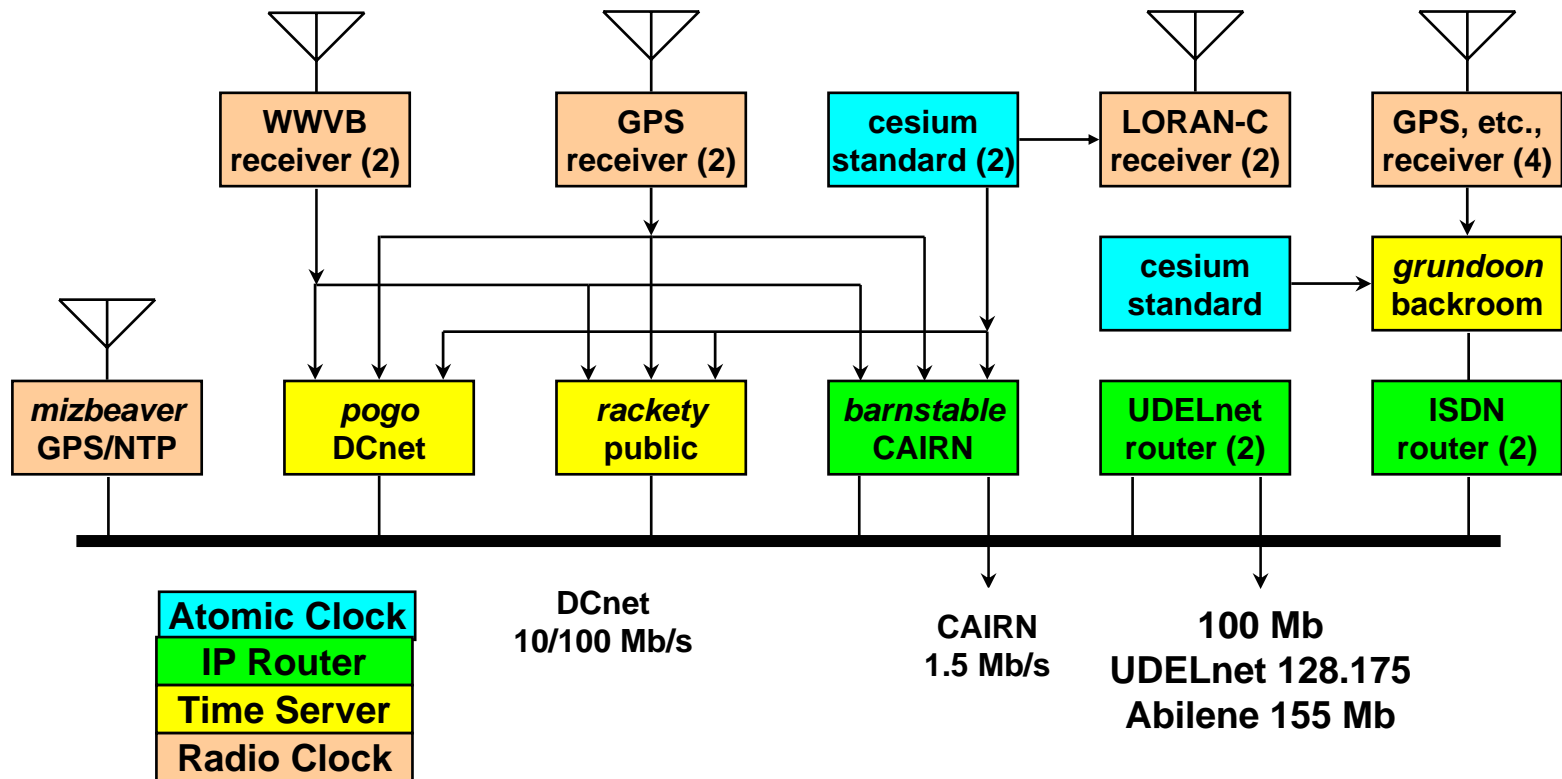
### **Beispiel: Hohe Präzision durch Kalkulation von Botschaftenlaufzeiten**

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'**



# DCnet timekeeping facilities (January 2001)





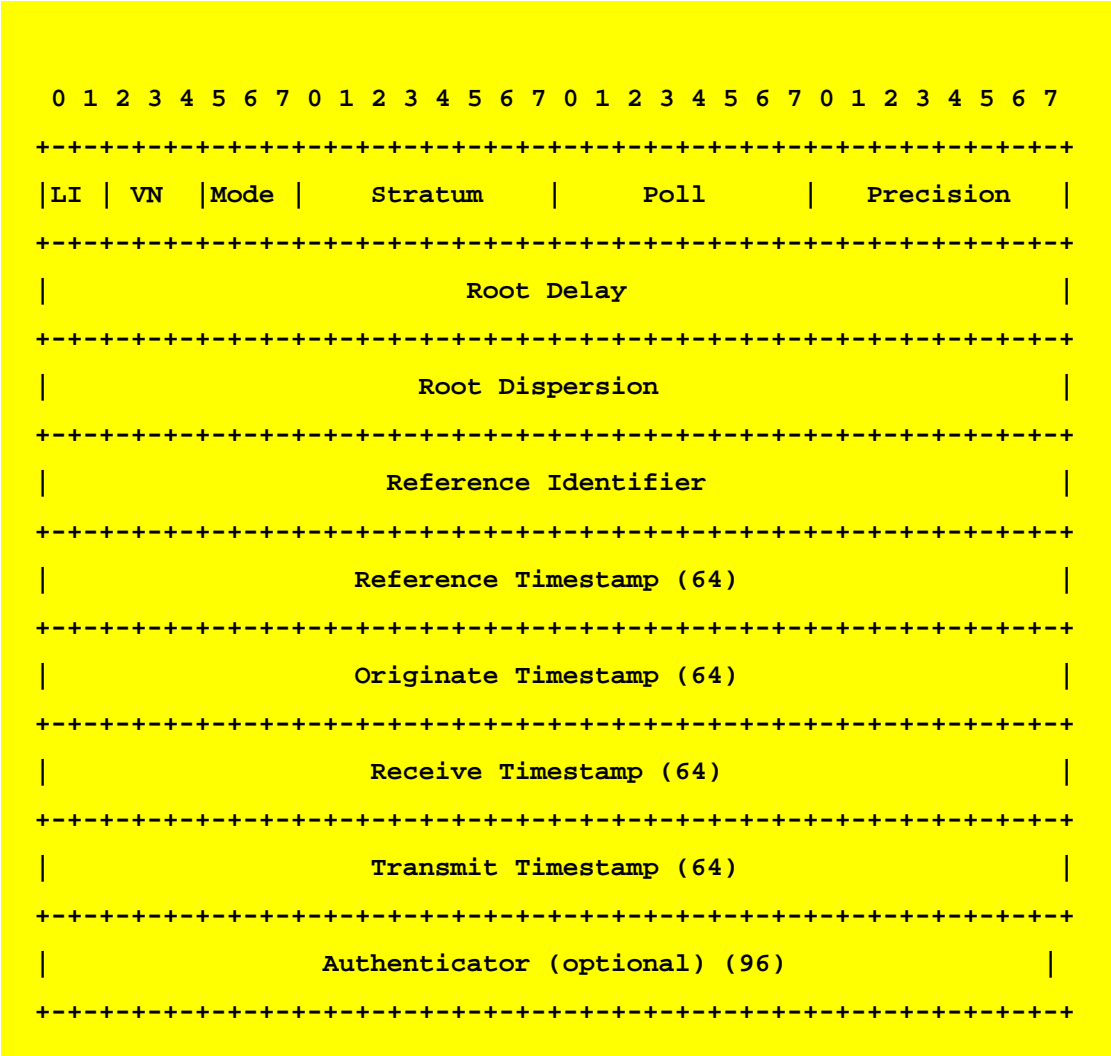
## **Weitere Maßnahmen zur Erhöhung der Genauigkeit**

**(precision + accuracy):**

-  **statistische Filteralgorithmen eliminieren Abweichungen von Botschaftenlaufzeiten**
-  **Gewichtung von Zeitinformationen nach ihrer Quelle (strati)**  
**=> Messungen: 99% aller Zeitinformationen weichen weniger als 30 msec von der realen UTC ab**



# NTP Packet Header



LI : Leap indicator, warnt vor einer anstehenden „Schaltsekunde“

VN: Versionsnummer der NTP Impl.

Mode: Operationsmodus, z.B. ob client oder Server

Stratum:  
 1: Primärreferenz (direkter Zugang zu einer ext. Zeitreferenz)  
 2-15: Sekundärreferenz  
 16-255: reserviert

Poll: Max. Intervall zwischen wie aufeinanderfolgenden Zeitnachrichten.  
 Angabe signed Int.: Exp. zur Basis 2, Einheit: Sek.

Precision: Präzision der lokalen Uhr  
 Angabe signed Int.: Exp. zur Basis 2, Einheit: Sek.

Root Delay: Roundtrip Delay

Root Dispersion: Fehler relativ zu einer Primärreferenz

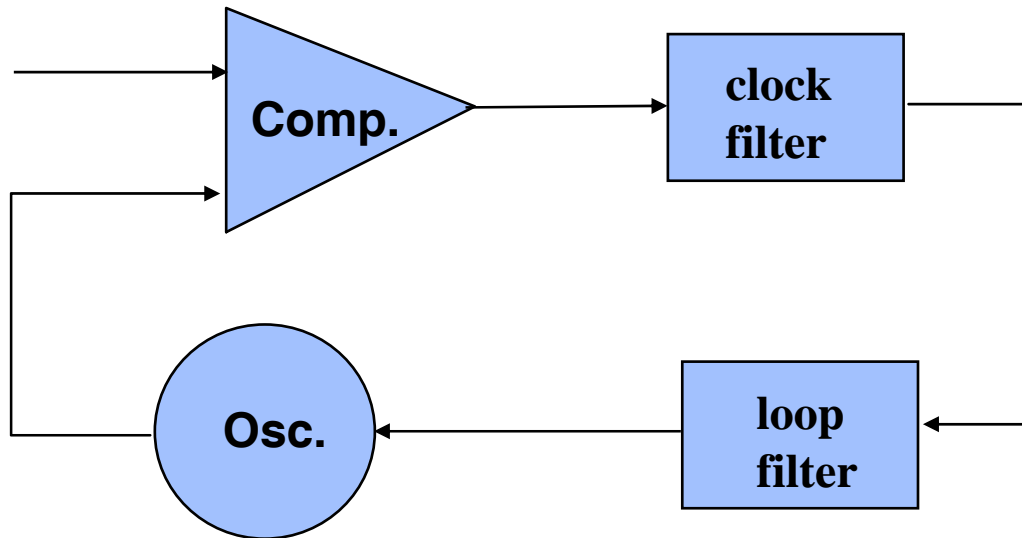
Ref. ID: ID der Referenzuhr (z.B. DCF77)

Ref.TS: Zeitpunkt des letzten Sync. Der Lokalen Uhr

Originate, Rx. Tx.: Client req., Server Rx. Sever Tx zur Roundtrip Berechnung.

Authenticator: für optionale Authentifizierung der Zeitstempel.

## PLL-Modell des NTP (RFC1129)

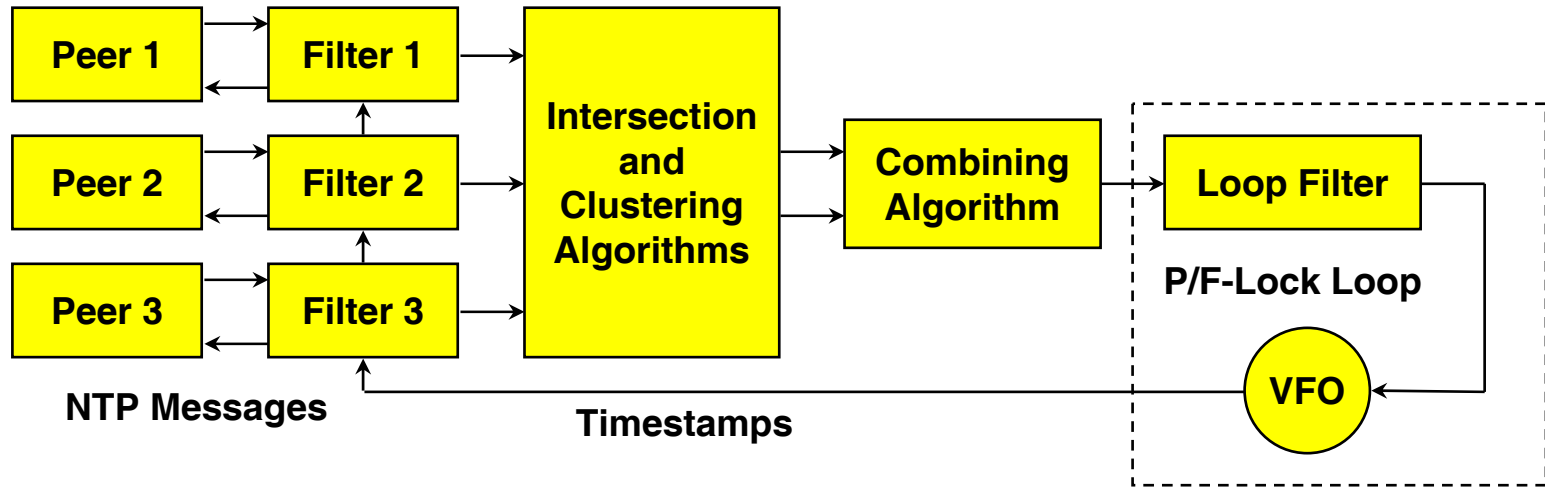


**NTP reguliert die Oszillatorfrequenz**

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

**loop filter:** berechnet den Korrekturwert für den Oszillator

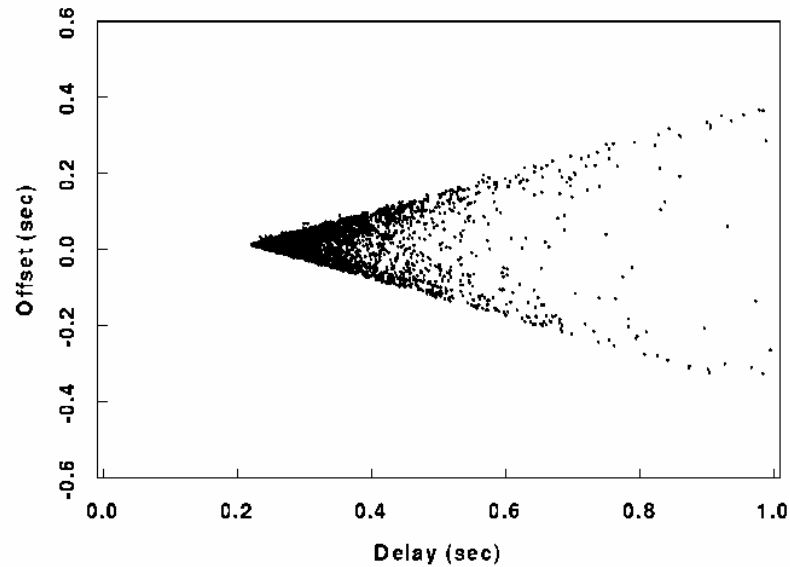
# How NTP works



- 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

## Messungen des Offsets

Offset  $o_i$



Roundtrip delay  $d_i$

**Clock Filter:**

- wählt Nachrichten aus, deren Verzögerung möglichst klein ist.

## 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.**

# Dezentrale und kooperative Verfahren zur Uhrensynchronisation

## **Averaging:**

L. Lamport, P. Melliar-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

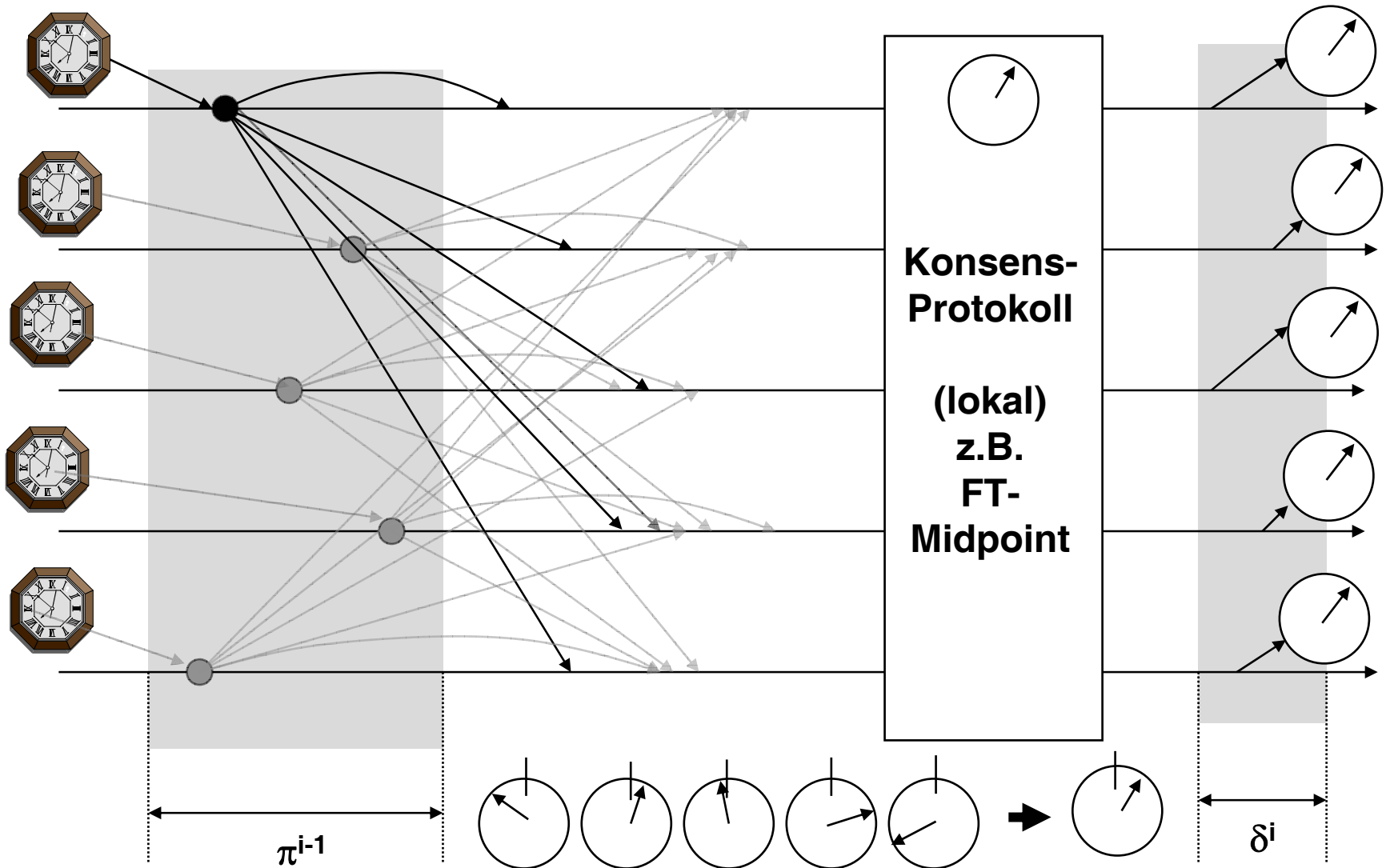
## **Hybrid:**

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



# Mittelwertes Verfahren zur Synchronisation (averaging)



$\delta$ : Verbesserung der „precision“  $\pi$

Bilden des Mittelwertes

## 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.

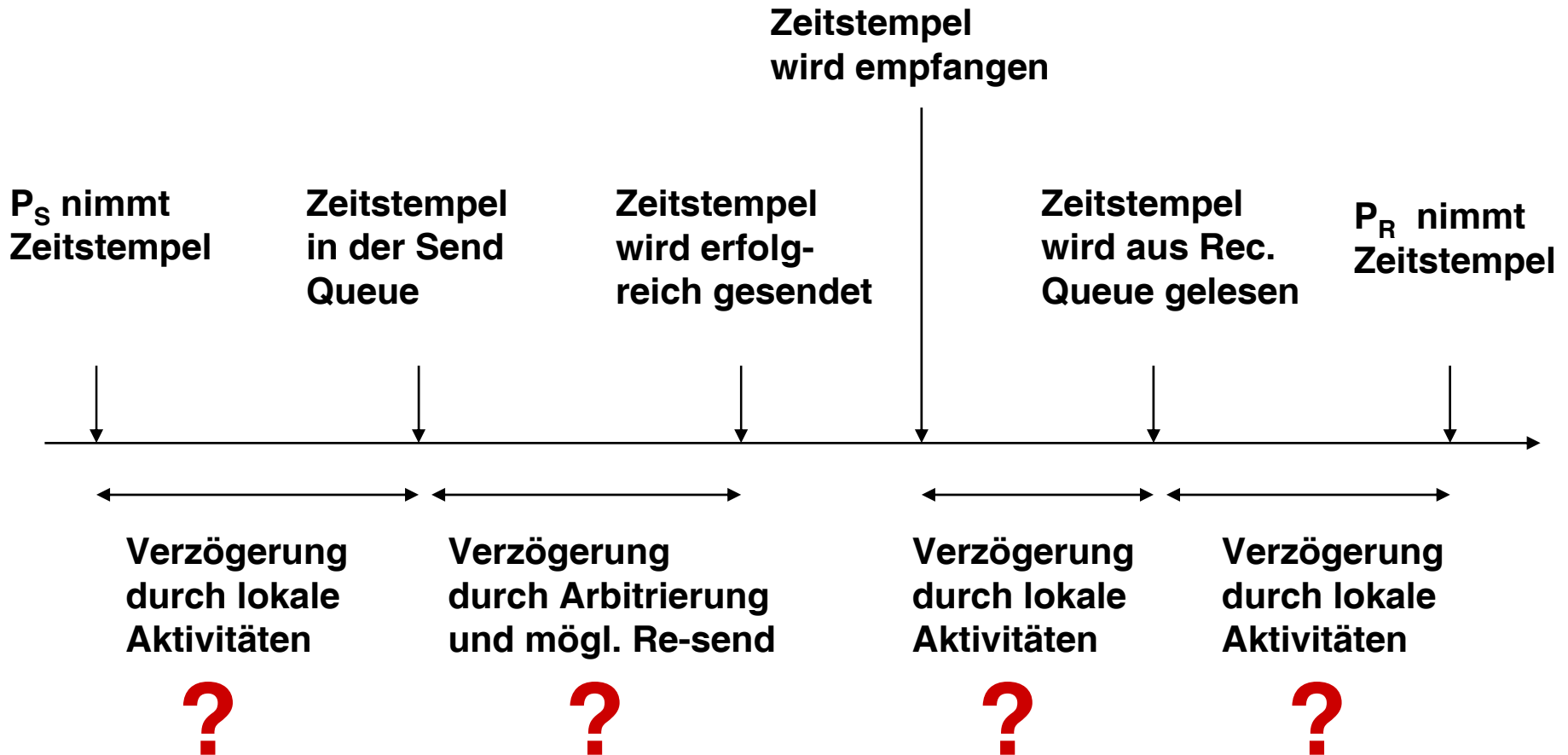
- Midpoint:  $(\text{max\_value} + \text{min\_value}) / 2$

- das ist **nicht** der Median in der sortierten Werteliste!!!!

### – Fault-tolerant Average:

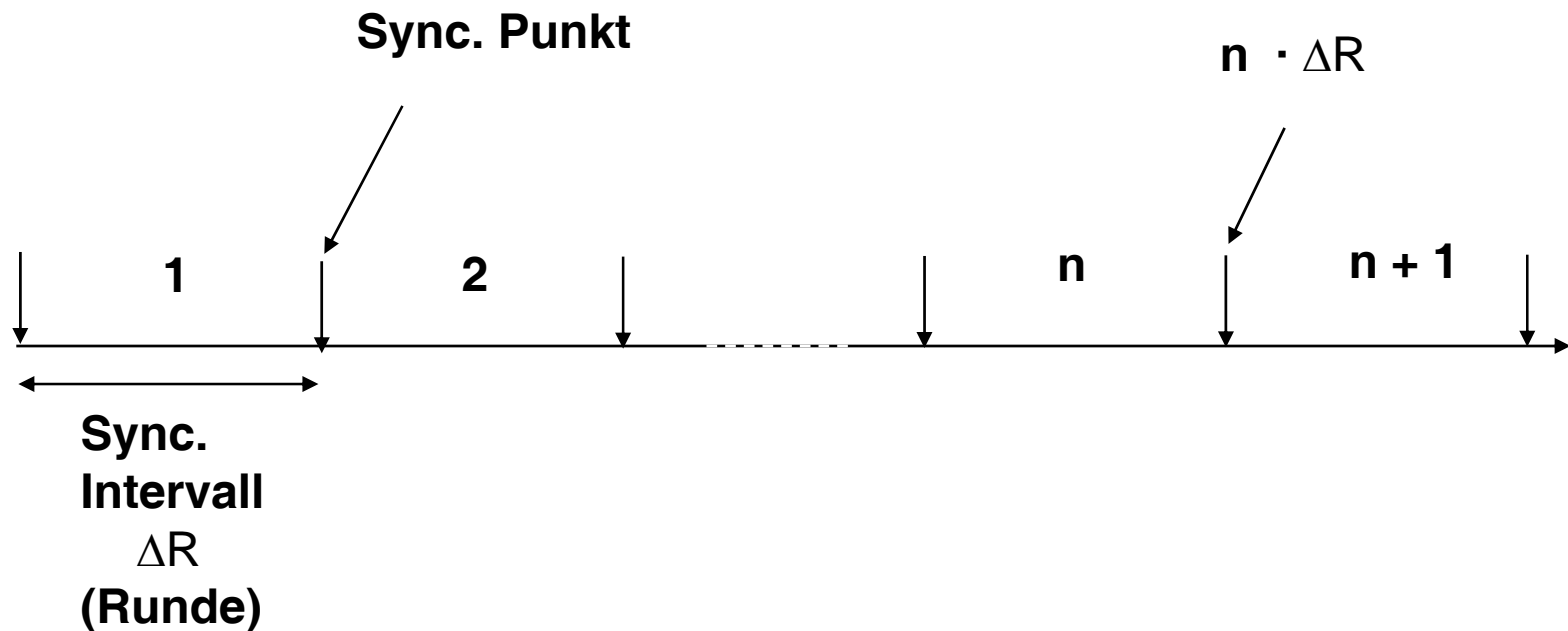
Die  $k$  höchsten and  $k$  niedrigsten Werte bleiben unberücksichtigt. Aus den verbleibenden Werten wird das arithmetische Mittel gebildet.

# Praktische Überlegungen:

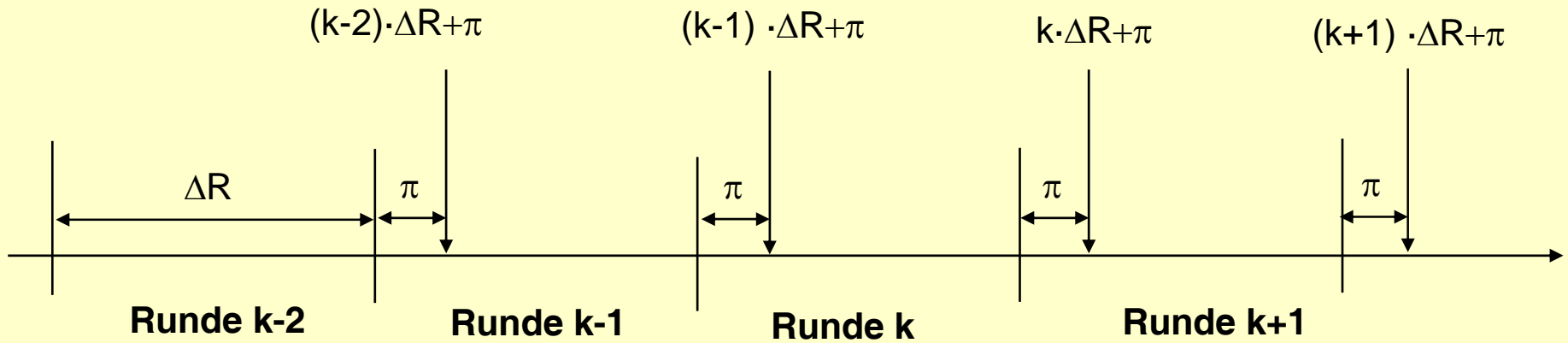


**Ausserdem: Viele zusätzliche Nachrichten erforderlich.**

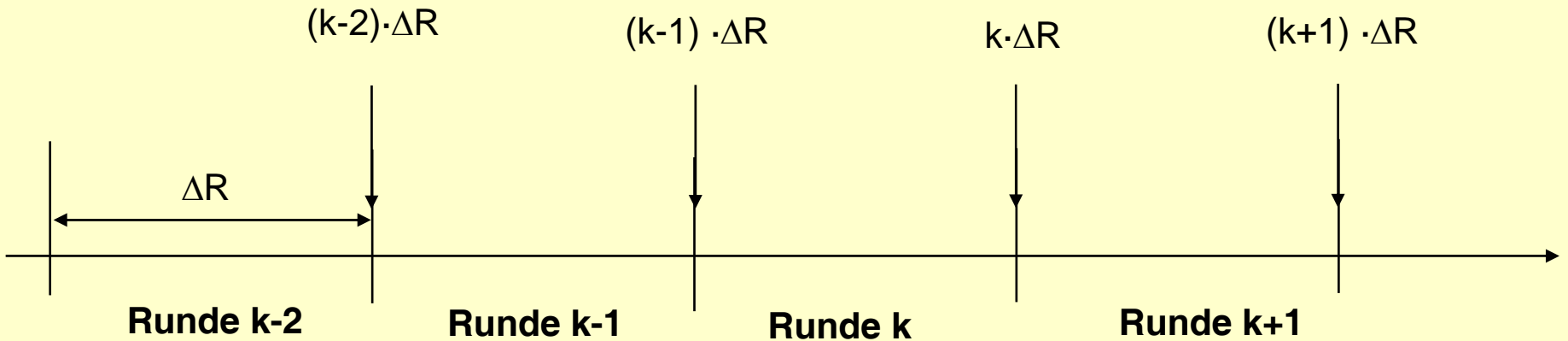
# Nicht-mittelndes Synchronisationsverfahren



# Synchronisation in einem nicht-mittelnden Verfahren

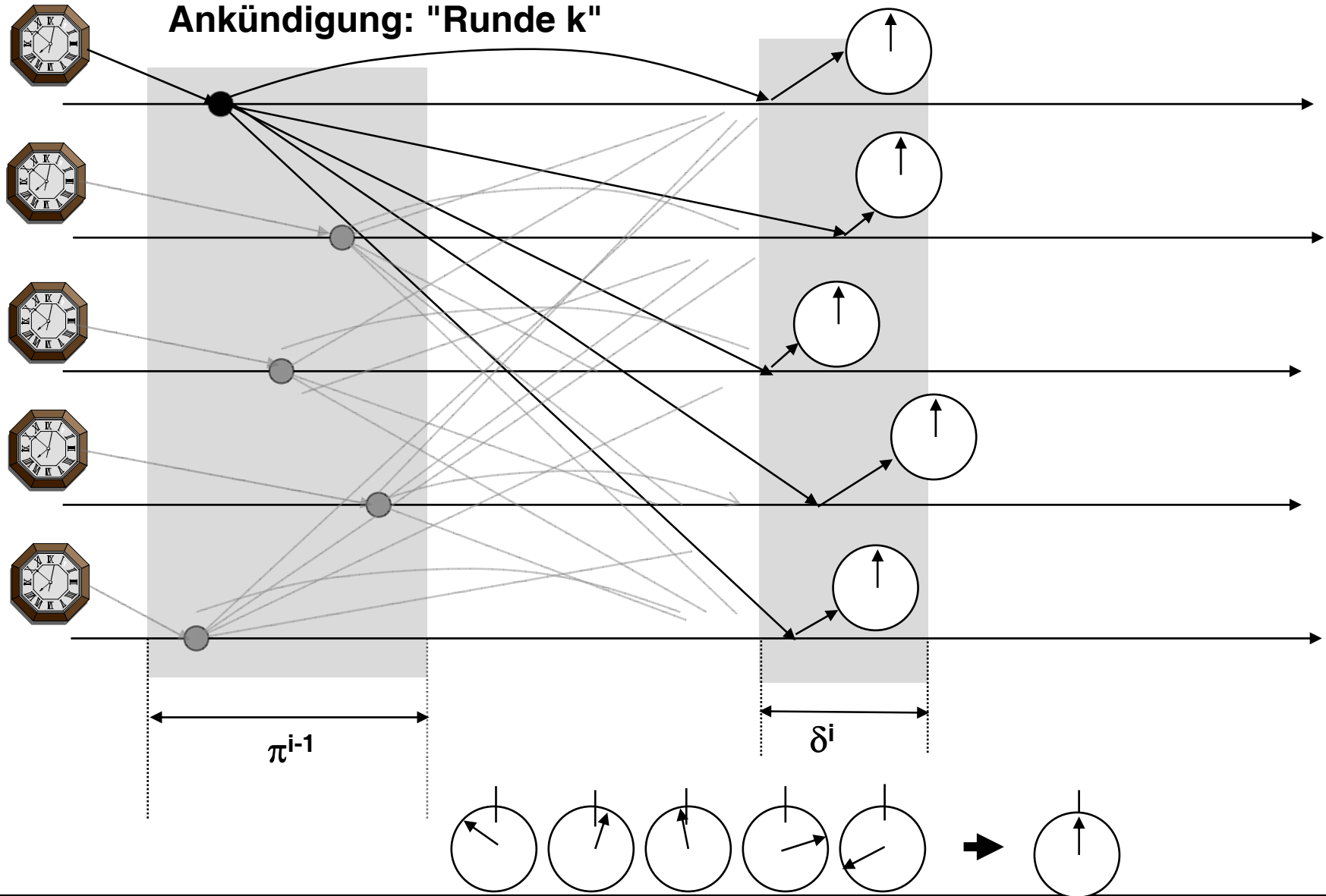


T. K. Srikanth and S. Toueg. Optimal clock synchronization. Journal of the ACM, 34(3):626645, July 1987.



[HSSD84] J. Y. Halpern, H. R. Strong, B. B. Simons, and D. Dolev. Fault-tolerant clock synchronization. In Proc. of 3rd International Symposium on Principles of Distributed Computing, pages 89102, 1984.

# Nicht mittelndes Verfahren (non averaging)

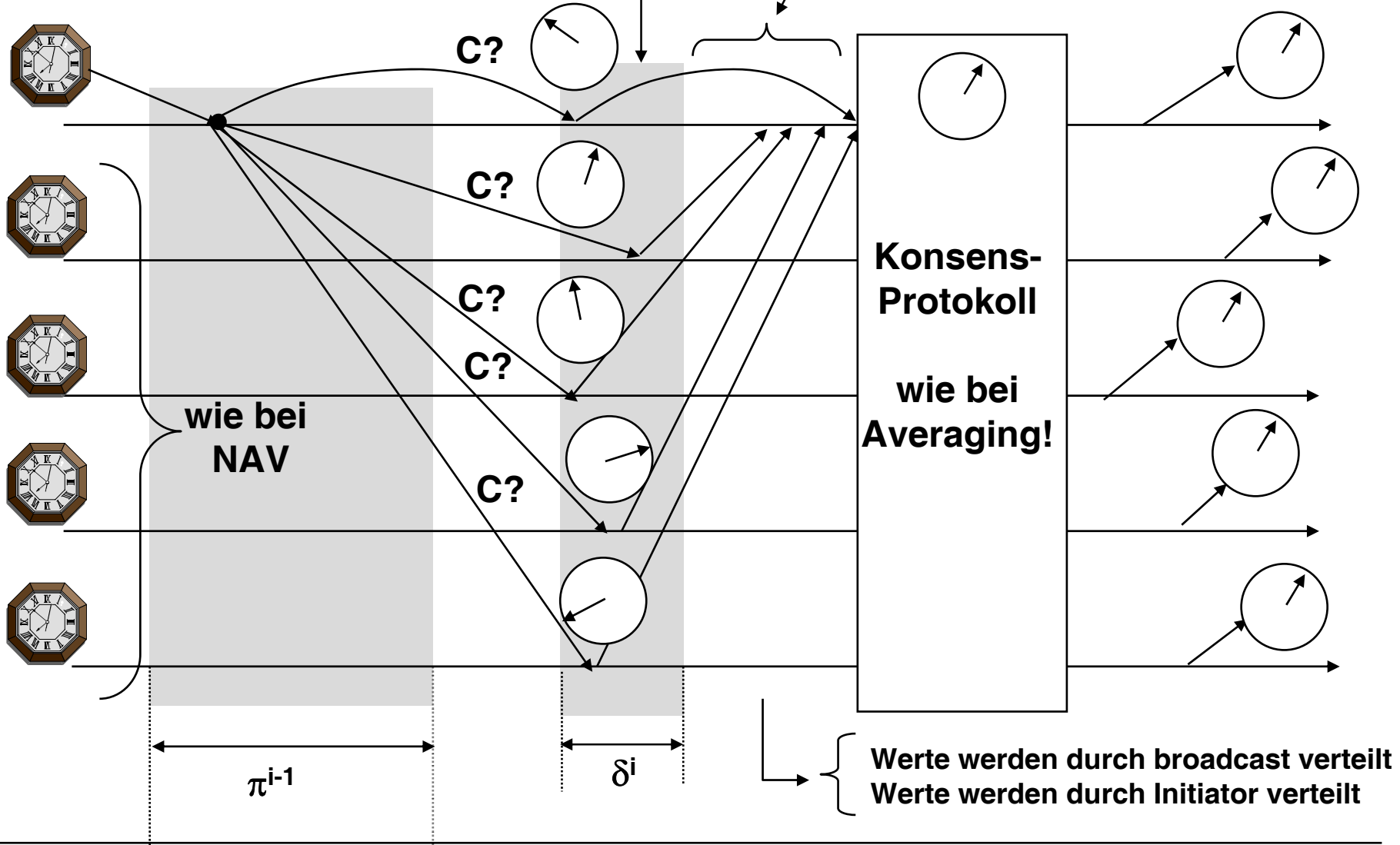


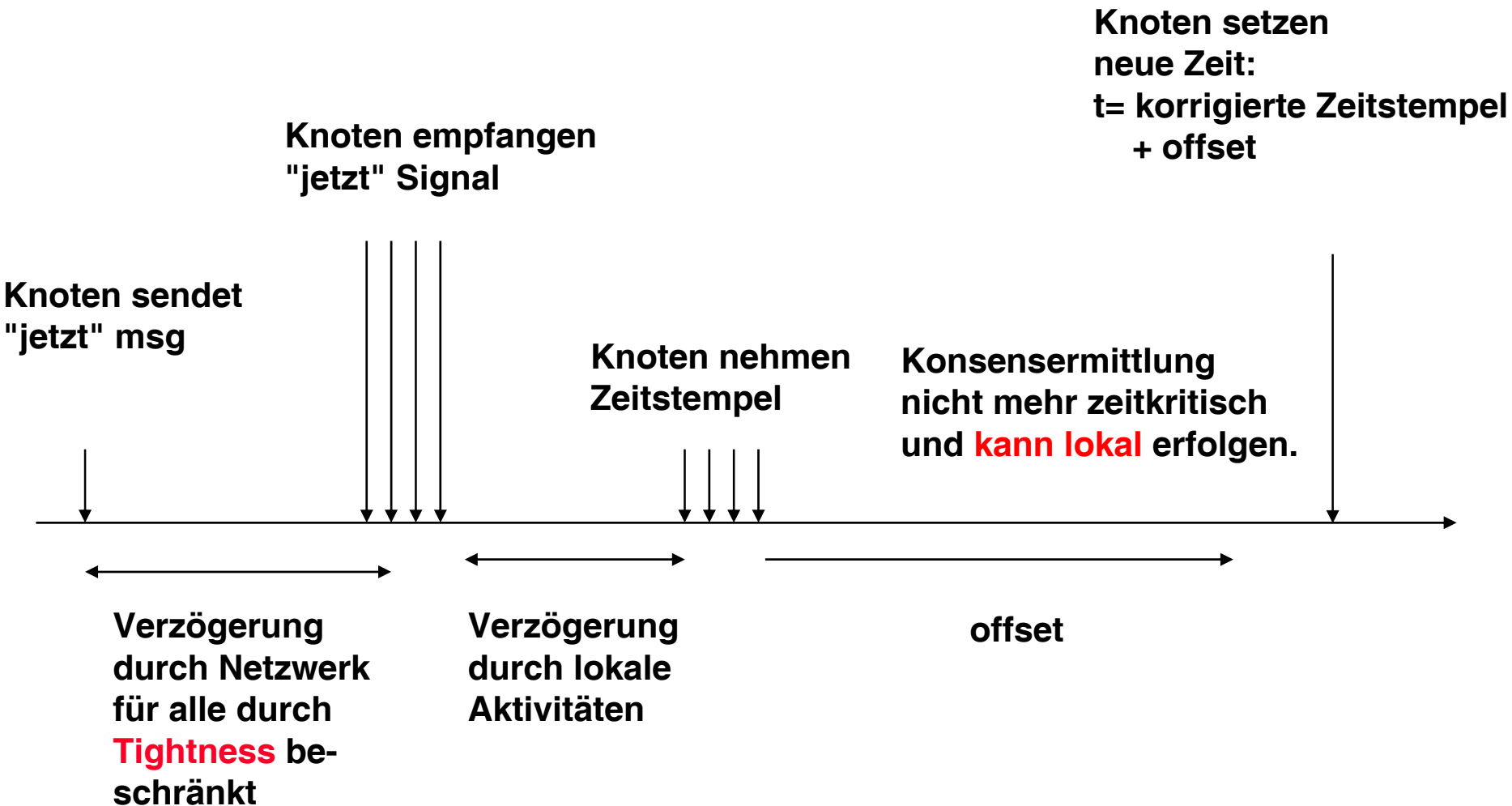
# Hybrides Verfahren

Uhr startet  
Sync. Protokoll

lokale Zeitstempel werden beim  
Empfang der Sync-N. genommen !

lokale Zeitstempel  
werden versendet







# 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 empfangenen Zeitwerte konstruieren als :

$$C_j^r = \begin{pmatrix} t^r_{1,1} & \cdot & \cdot & \cdot & \cdot & t^r_{1,j} & \cdot & \cdot & \cdot & \cdot \\ \cdot & t^r_{2,2} & \cdot & \cdot & \cdot & t^r_{2,j} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & t^r_{3,j} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & t^r_{j,j} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & t^r_{n,j} & \cdot & \cdot & \cdot & t^r_{n,n} \end{pmatrix}$$

\*

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

Die Diagonale enthält die Zeitstempel der Nachricht von Rechner i,j (i=j)  
die Spalte enthält die Empfangszeiten der Nachricht auf den übrigen Rechnern.

Für alle “guten” Uhren gilt:

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

$O_{i,j}$  (offset) : Abweichung der Uhr auf Rechner i von der Uhr auf Rechner j

md : Verzögerung der Nachricht (angenommen)

$Er_{i,j}$  : Zusammenfassung der Auslesefehler auf Rechner i und Rechner j

# Kooperierende Zeitsynchronisationsverfahren

Korrekturvektor für Rechner j

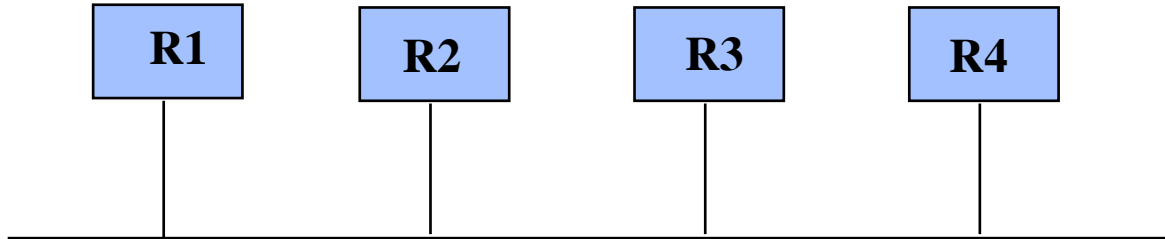
$$\mathbf{K}^t = \left( \begin{array}{ccccccc}
 \mathbf{0} & k_{1,2} & \dots & k_{1,j} & \dots & k_{1,n} \\
 k_{2,1} & \mathbf{0} & \dots & k_{2,j} & \dots & k_{2,n} \\
 \cdot & \cdot & & k_{3,j} & & \cdot \\
 \cdot & \cdot & \mathbf{0} & & & \cdot \\
 \cdot & \cdot & \dots & \mathbf{0} & \dots & \cdot \\
 \cdot & \cdot & & & \mathbf{0} & \cdot \\
 k_{n,1} & k_{n,2} & \dots & k_{n,j} & \dots & \mathbf{0}
 \end{array} \right)$$

Für jedes Element  $k_{i,j}$  gilt:

$$k_{i,j} = t_{i,j}^r - t_{i,i}^r - md \cdot N \quad ( N=1 \text{ wenn } i \neq j, N=0 \text{ wenn } i=j )$$

# Kooperierende Zeitsynchronisationsverfahren

Beispiel:



**md=6**

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

$C_j^r =$

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**

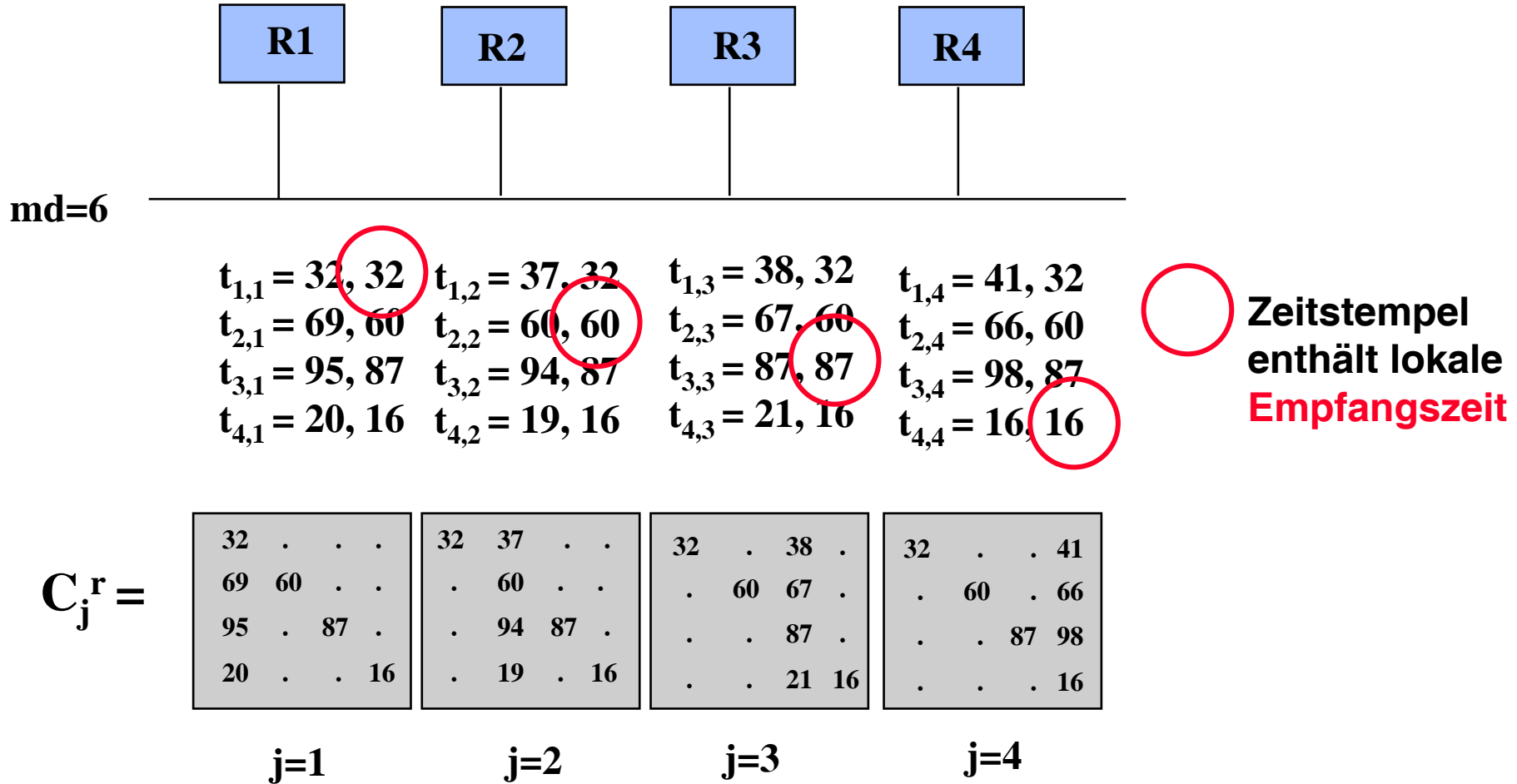
**j=2**

**j=3**

**j=4**

# Kooperierende Zeitsynchronisationsverfahren

Beispiel:



# Kooperierende Zeitsynchronisationsverfahren

$C_j^r =$

32	.	.	.	32	38	.	.	32	.	38	.	32	.	.	41
69	60	.	.	.	60	.	.	.	60	67	.	.	60	.	71
95	.	87	.	.	94	87	.	.	.	87	.	.	.	87	98
20	.	.	16	.	19	.	16	.	.	21	16	.	.	.	16

$j = 1$

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

$j = 2$

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

$j = 3$

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

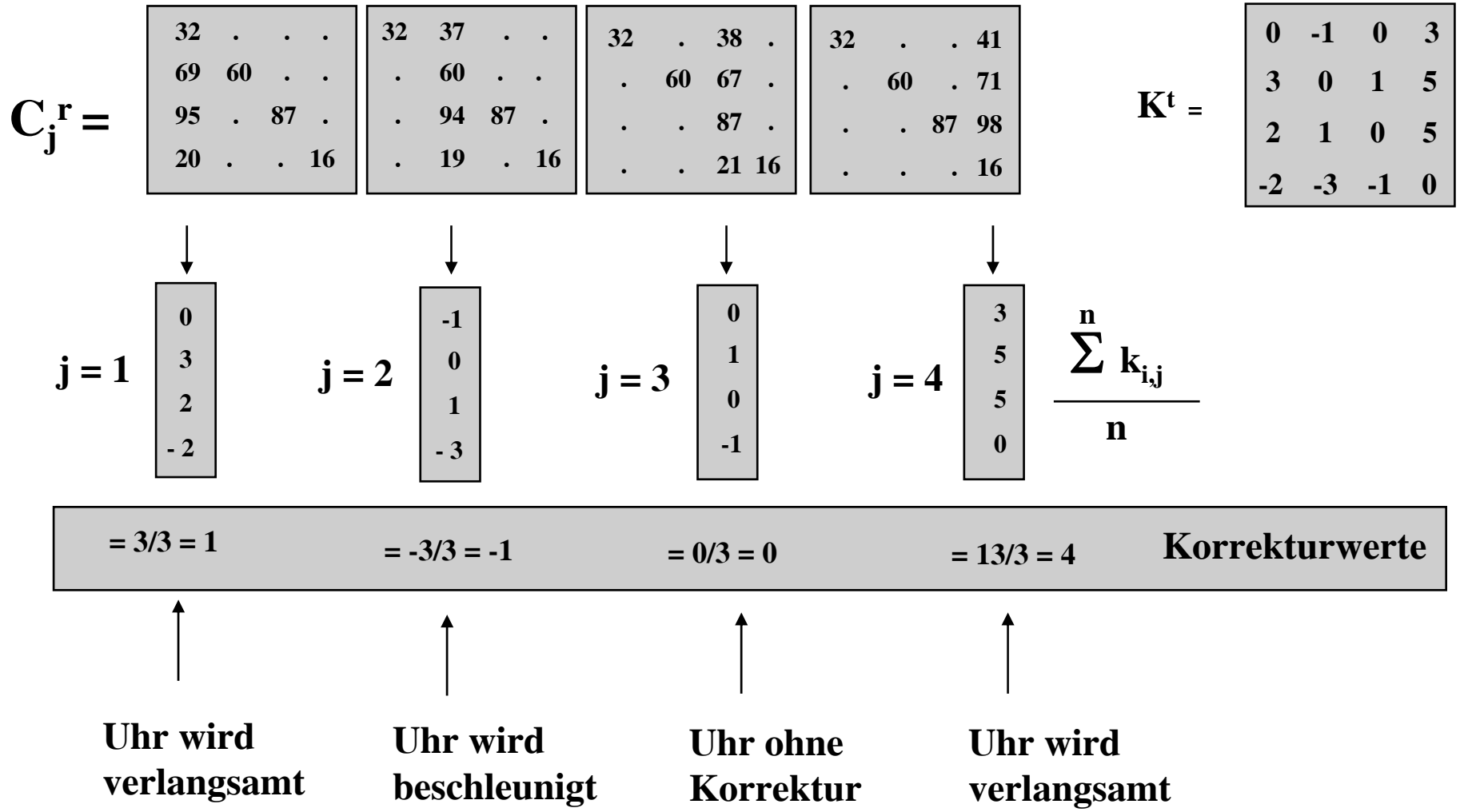
$j = 4$

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

$K^t =$

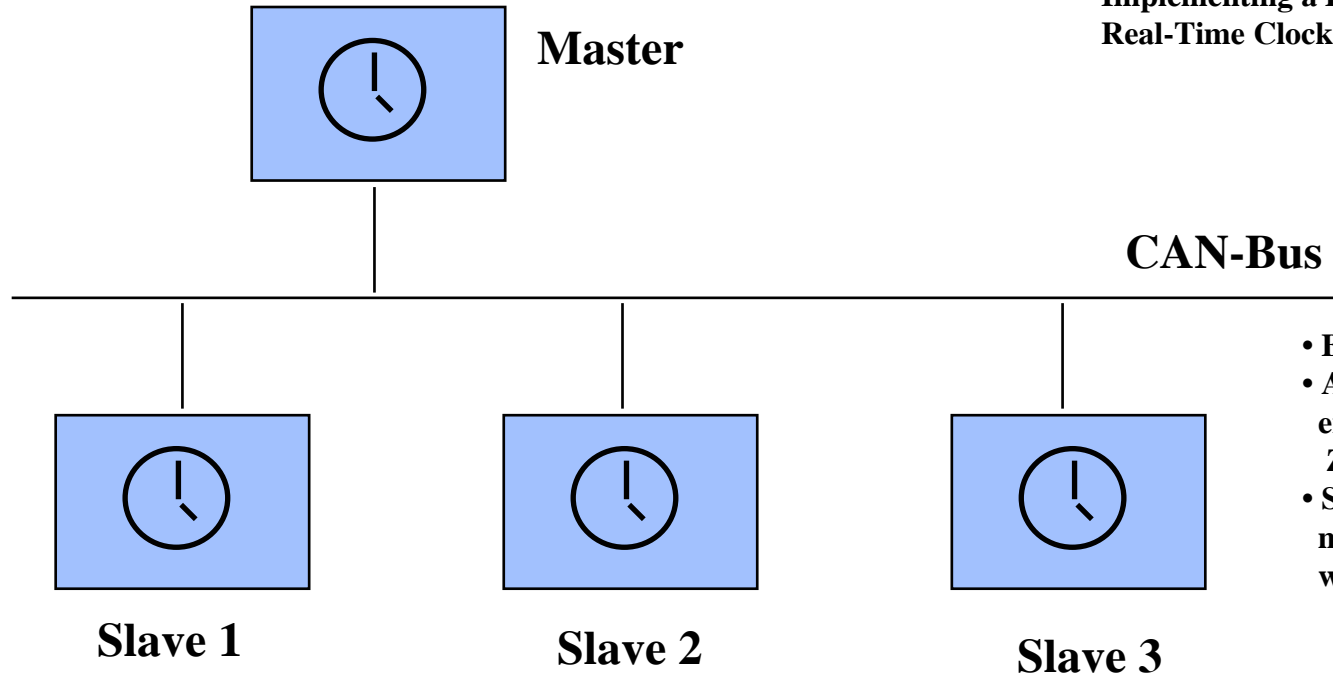
0	-1	0	3
3	0	1	5
2	1	0	5
-2	-3	-1	0

# Kooperierende Zeitsynchronisationsverfahren



# 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

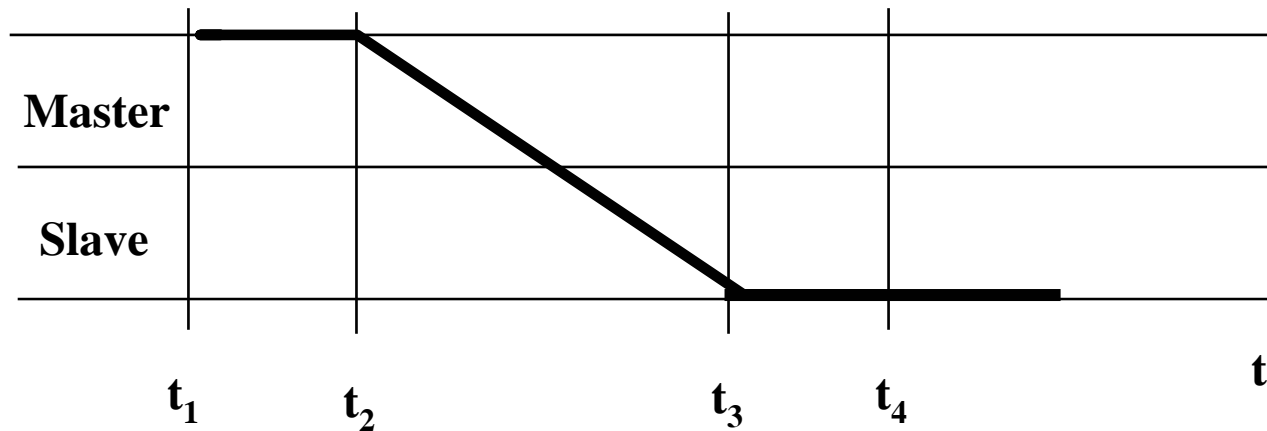
## System:

- 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**

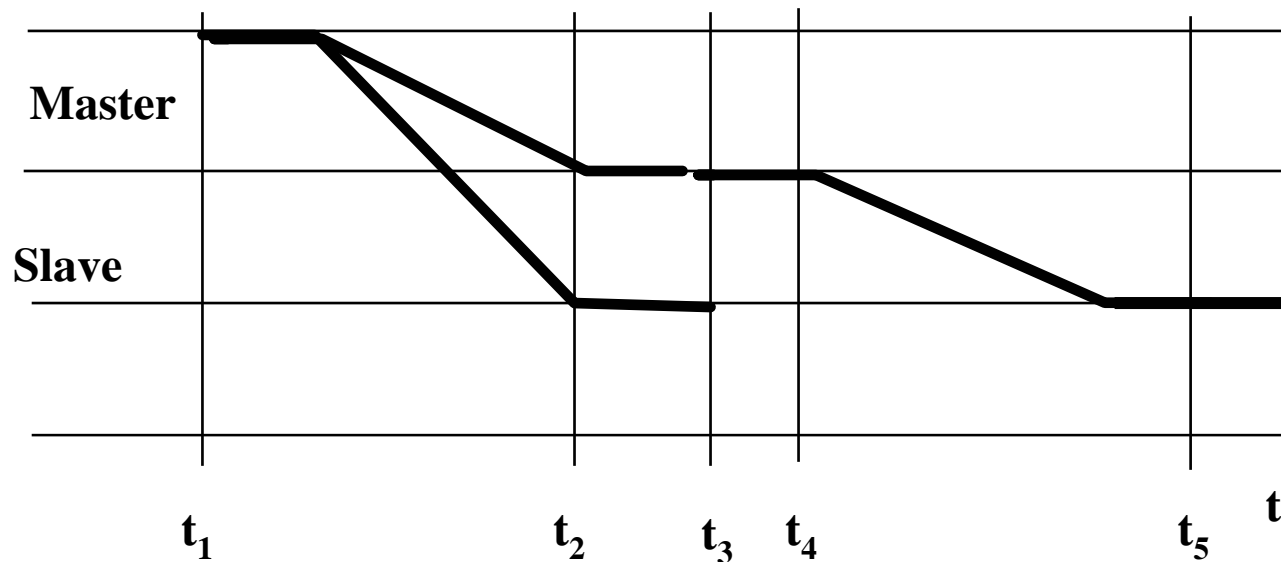




# Ein Echtzeitsynchronisationsprotokoll für den CAN-Bus

## Verbessertes Protokoll

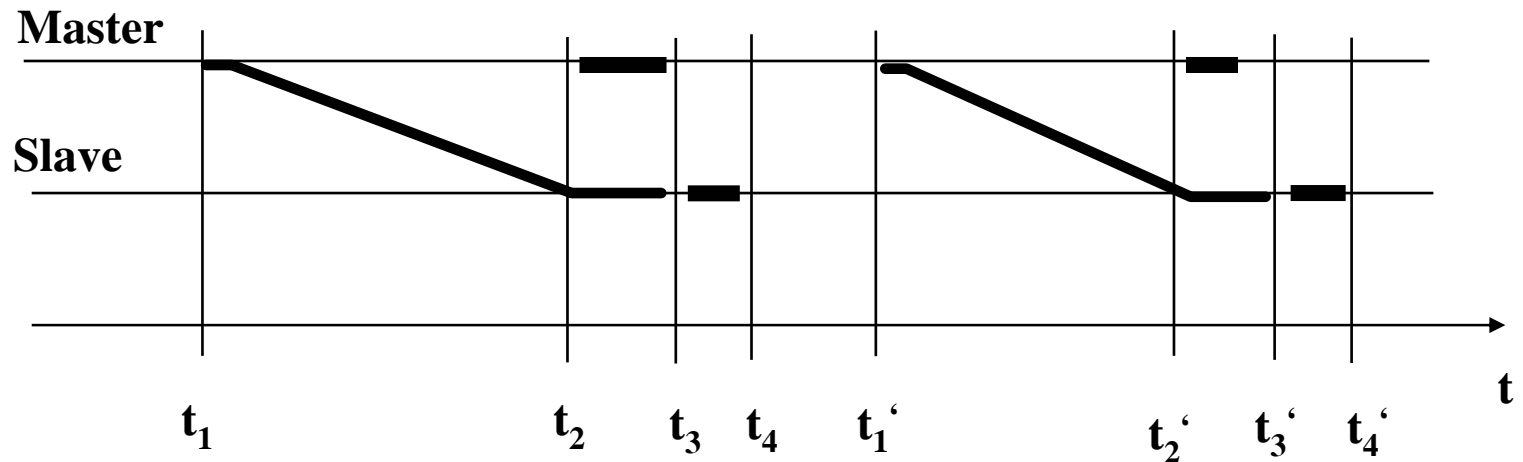
- Irgendein Rechner  $i$  sendet die Nachricht "jetzt" ( $t_1$ )
- Master und Slaves lesen ihre lokale Zeit beim Empfang der Nachricht ( $t_2$ )
- Master sendet seinen Zeitstempel ( $t_3, t_4$ )
- Slaves gleichen ihre Zeit danach ab ( $t_5$ )



# Ein Echtzeitsynchronisationsprotokoll für den CAN-Bus

## Optimiertes Protokoll

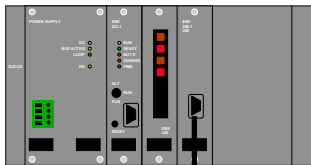
- **Master sendet die Nachricht "jetzt" ( $t_1$ )**
- **Die Nachricht enthält den Zeitstempel der letzten Sync.-Runde**
- **Master liest seine Uhrzeit, beim Empfangs-Interrupt ( $t_3$ )**
- **Slaves lesen ihre Uhrzeit, beim Empfangs-Interrupt ( $t_3$ )**
- **Slaves gleichen ihre Zeit nach dem letzten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



# Ein Echtzeitsynchronisationsprotokoll für den CAN-Bus

- ◎ **mittlere Genauigkeit**
  - etwa 30 usec
- ◎ **Anschluß an externe Zeit**
  - GPS Satellitenempfänger
- ◎ **keine zusätzlichen Kosten**
  - implementiert auf embedded Controllern und CAN-Feldbus
- ◎ **Anwendungsbeispiele:**
  - verteilte Meßdatenerfassung
  - verteilte kooperierende Aktoren

11:58 AM



11:58 AM

