Uhrensynchronisation: Untergliederung

5 **Uhrensynchronisation**

5.1 **Notwendigkeit der Uhrensynchronisation, logische Uhren, physikalische Uhren**

5.2 **Konstruktion logischer Uhren**

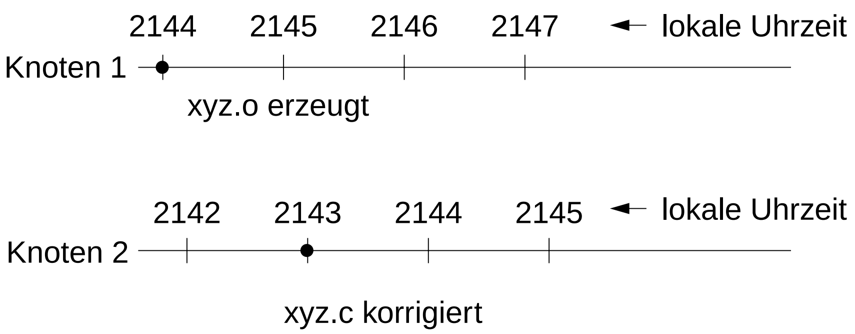
5.3 **Synchronisation physikalischer Uhren**

Uhrensynchronisation: Logische und physikalische Uhren

5.1 **Notwendigkeit der Uhrensynchronisation, logische Uhren, physikalische Uhren**

❑ **Zeit als Mittel der Reihenfolgebestimmung**

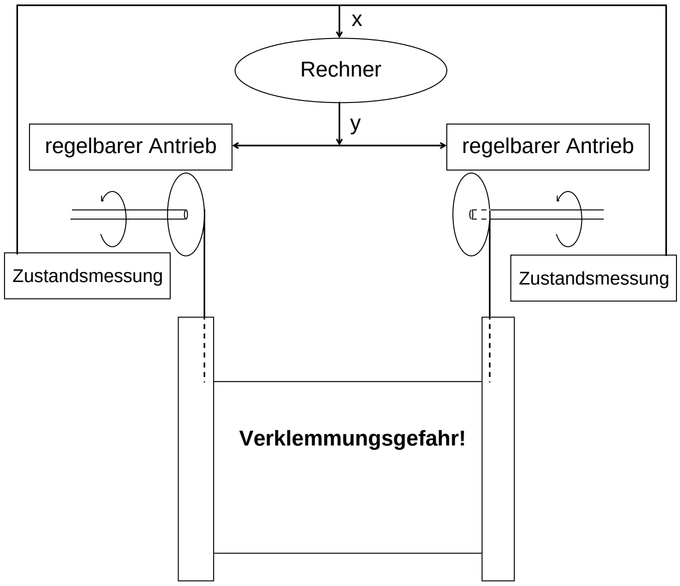
- **Beispiel 1: make**



- **Beispiel 2: P, V**
  - **Semaphor-Replikat auf jedem Knoten**
  - **Operationsaufruf wird mit Zeitstempel versehen und per Broadcast verschickt**
  - **Jeder Knoten arbeitet die Aufrufe lokal nach Zeitstempeln geordnet ab (verzögert um maximale Übertragungszeit)**

❏ Zeit als Mittel zur Ausgabesynchronisation

- **Beispiel: Koordiniertes Heben**



❏ Synchronisationsarten

- **schwache Synchronisation**
  - **logische Uhren**
    - **Synchronisation bei Kommunikation**
- **starke Synchronisation**
  - **regelmäßige Nachführung**
    - **Master-Slave mit Verteilung einer Referenzzeit**

      Broadcasting (Problem: Laufzeit)

      Polling (z. B. mit rdate; Auflösung im Sekundenbereich, Laufzeitprobleme)
    - **Einigungsverfahren: alle zeigen gleiche Zeit, die sich nicht unbedingt mit der amtlichen Zeit deckt**

      DTS - Distributed Time Service

      NTP - Network Time Protocol (Public Domain)

BP 1 **Uhrensynchronisation: Logische und physikalische Uhren**

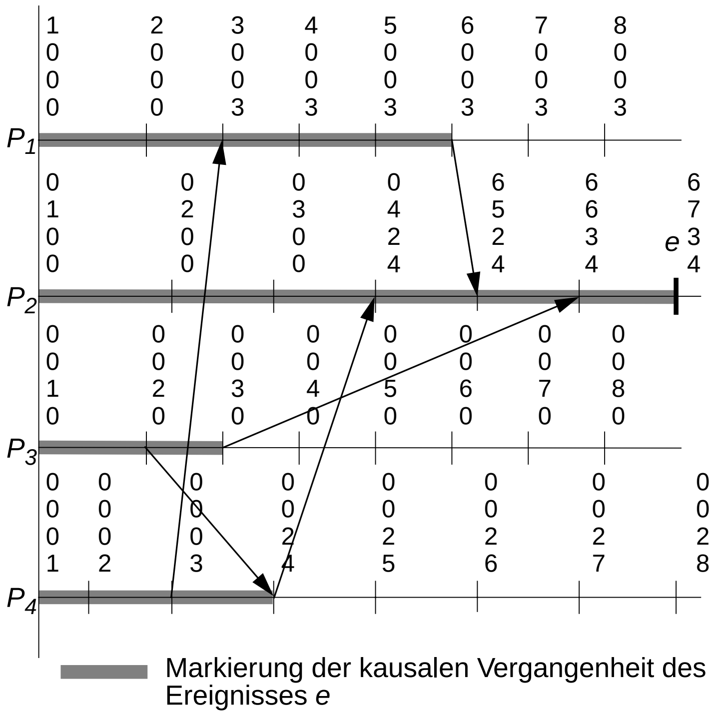
❑ **Bewertungskriterien**

- **Monotonie**
- **Synchronisation mit der amtlichen Zeit**
- **Robustheit gegen Netzpartitionierung**
- **Fehlertoleranz gegen falsch gehende Referenzuhren**
- **Referenztreue**
- **Erzeugte Netzlast**
- **Fehleraussage**
  Zeitdifferenz in LAN's: kleiner als 2 ms

BP 1 **Uhrensynchronisation: Vektorzeit**

5.2 **Logische Uhren**

Vektorzeit



5.3 Physikalische Uhren

◆ Grundbegriffe

- **Realzeit *t*: Eine physikalisch definierte, von den Prozessen nicht unmittelbar beobachtbare Größe.**
- **Uhrzeit *T*: Zeit, die an einer bestimmten Uhr abgelesen werden kann.**
- **Lokale Uhr *C* zeigt zur Realzeit *t* den Wert *T*:**

  $C(t) = T$ **, wobei *C* streng monoton wachsend ist.**

  Die inverse Funktion wird mit c bezeichnet, also $c(T) = C^{-1}(T)$

- **Ablesen der Uhr *C* liefert zur Realzeit *t* die Uhrzeit UC(t) .**

  $UC(t) = aC(t) + b$ **, wobei *a* und *b* diskontinuierlich verändert werden, um Synchronisation zwischen** $UC(t)$ **und der Realzeit zu erreichen.**

- **Intervalluhr *IC* (definiert durch IC_u und IC_o) liefert Intervall, das die Realzeit *t* enthält:**

  $t \in IC(t) = [IC_u(t), IC_o(t)]$ **, wobei IC_u und IC_o monoton wachsend sind.**

◆ Physikalische Zeit basierend auf der Atom-Sekunde

- **Die Sekunde ist das 9192631770-fache der Periodendauer der dem Übergang zwischen den beiden Hyper-Feinstruktur-Niveaus des Grundzustandes von Atomen des Nuklids ^{133}Cs entsprechenden Strahlung.**
  TAI Temps Atomique International (seit 1972)

  Genauigkeit von Cäsium-Uhren: ca. 10^{-13}, d. h. 8,6 ns/Tag oder rund acht

  Taktzeiten eines heutigen Hochleistungsrechners

◆ UTC Universal Time Coordinated (seit 1972 als Nachfolger der Greenwich Mean Time), basierend auf internationalen astronomischen Beobachtungen; UTC gegenwärtig 1 s/Jahr schneller als TAI, Korrektur durch Einfügung von Schaltsekunden.

- Nationale amtliche Zeit UTC + nationale Festlegungen der Differenz zu UTC

Eigenschaften:

- Kein Zeitpunkt kommt mehrfach vor.
- Verbreitung der amtlichen Zeit durch satellitenbasierte Systeme (z. B. Global Positioning System GPS und Geosynchronous Orbit Environmental Satellites GOES) und terrestrische Sender (z. B. das Signal DCF77 von Mainflingen aus durch einen Sender der Physikalisch Technischen Bundesanstalt mit einer erreichbaren Zeitauflösung von ca. 10 μs).
- Ausgesandt werden Normalfrequenz und Zeitzeichen.

Synchronisationsalgorithmen

- Annahmen

A0 Alle Uhren sind anfänglich auf etwa die gleiche Uhrzeit synchronisiert

A1 Die Uhren fehlerfreier Prozesse gehen annähernd richtig, d. h. sie besitzen hohe Ganggenauigkeit.

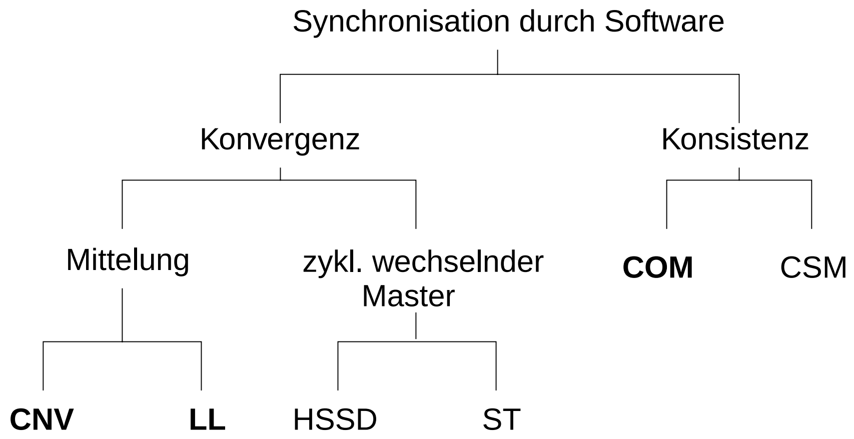
A2 Ein fehlerfreier Prozeß p kann die Differenz δ_{pq} zwischen seiner Uhr und der von q mit einer Genauigkeit ε lesen.

- Wunschvorstellung

S1 Zu jedem Zeitpunkt zeigen die Uhren zweier fehlerfreier Prozesse ungefähr die gleiche Zeit

S2 Es gibt eine kleine obere Schranke für den Wert, um den die Uhr eines fehlerfreien Prozesses bei einer Resynchronisation verstellt wird.

Uhrensynchronisation: Physikalische Zeit



ST Srikanth, T.K., and S. Toueg. Optimal clock synchronization. JACM 34, 3 (July 1987), 626-645.

HSSD Halpern, J.Y., B. Simons, R. Strong and D. Dolev. Fault-tolerant clock synchronization. Proc. ACM Third Annual Symposium on Principles of Distributed Computing (August 1984), 89-102.

Uhrensynchronisation: Physikalische Zeit

- **Bezeichnungen**

γ	**Maximale Abweichung zwischen den verschiedenen Uhren**
δ_0	**Maximale Abweichung am Anfang**
ε	**Maximaler Fehler beim Ablesen der Uhren**
δ	**"Normale" Nachrichtenübertragungszeit (Nachrichtenlaufzeit $\delta \pm \varepsilon$)**
β	**Maximale Abweichung der Realzeit der Uhren im Anfangszustand T^0**
ρ	**Relative Abweichung der Taktrate**
$\Delta_{q,p}$	**Differenz zwischen Uhr von q und Uhr von p so, wie sie p durch Ablesen der Uhr von q feststellt**
Σ	**Maximale Verstellung von Uhren**
m	**Maximalzahl fehlerhafter Prozesse**
n	**Gesamtzahl der Prozesse**
R	**Resynchronisationsintervall**
$R^{(i)}$	**i-tes Synchronisationsintervall**
$S^{(i)}$	**Intervall zur Durchführung des Synchronisationsalgorithmus am Ende des i-ten Synchronisationsintervalls**
$T^{(i)}$	**Ende der i-ten Synchronisationsrunde**

5.3.1 Konvergenz-Algorithmus CNV

Literatur:

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

- **Jeder Prozeß liest die Uhr jedes anderen und stellt seine eigene auf den Mittelwert. Dabei wird für Uhren die mehr als γ von der eigenen abweichen, der Wert der eigenen Uhr genommen.**

Dieser Algorithmus besitzt die gewünschten Eigenschaften, wenn $n > 3m$ ist.

Plausibilitätsbetrachtung:

p, q **seien fehlerfrei,** r **irgendein Prozeß.**

$c_{p,q}$ **sei die Uhrzeit von** q**, so wie sie** p **bekannt ist.**

r **fehlerfrei** $\Rightarrow c_{p,r} \approx c_{q,r}$

r **fehlerhaft** $\Rightarrow |c_{p,r} - c_{q,r}| \le 3\gamma$

denn

$$|c_{p,r} - c_{p,p}| \le \gamma \wedge |c_{q,r} - c_{q,q}| \le \gamma \wedge |c_{p,p} - c_{q,q}| \le \gamma$$

$$\Rightarrow |c_{p,r} - c_{q,r}| \le 3\gamma$$

Uhrensynchronisation: CNV

p stellt Uhr auf $\frac{1}{n}\sum_{r} c_{p,r}$

q stellt Uhr auf $\frac{1}{n}\sum_{r} c_{q,r}$

schlechtester Fall: (n-m)-mal $c_{p,r} \approx c_{q,r}$

m-mal $|c_{p,r} - c_{q,r}| \leq 3\gamma$

$$\Rightarrow \frac{1}{n}\left|\sum_{r} c_{p,r} - \sum_{r} c_{q,r}\right| \leq \frac{m}{n} 3\gamma < \gamma$$

Vernachlässigt sind dabei:

- die Zeit für die Ausführung des Algorithmus
- Fehler durch nicht gleichzeitiges Ablesen der Uhren

Uhrensynchronisation: LL

5.3.2 Konvergenz-Algorithmus LL

Literatur:

Lundelius Welch, J.; Lynch, N.: A New Fault-Tolerant Algorithm for Clock Synchronization. Information and Computation, Vol. 77 (1988), pp. 1-36.

◆ Prozesse, Uhren, System

- Ein verteiltes System wird modelliert als eine Menge von Prozessen, die über Nachrichten kommunizieren.
- Prozesse sind unterbrechungsgetrieben.
- Unterbrechungen werden verursacht durch Nachrichten, Startsignal (modelliert als Start-Nachricht START) oder Ablauf eines vorgebbaren Zeitintervalls (modelliert als Zeitgeber-Nachricht TIMER).

◆ Nachrichtensystem

- Jeder Prozeß kann jedem anderen unmittelbar eine Nachricht zusenden.
- Das Kommunikationssystem wird modelliert als globaler Nachrichtenpuffer.
- Eine Nachricht besteht aus der Sendezeit t und der Empfangszeit t'.
- Anfänglich enthält der Puffer für jeden Prozeß genau eine Start-Nachricht.

Uhrensynchronisation: LL

- **Algorithmus**
  - **Globale Konstante**

n	**Anzahl aller Prozesse (Prozessoren)**
f	**maximale Zahl fehlerhafter Prozesse**
ρ	**maximale Frequenzabweichung**
β	**maximale Abweichung beim Rundenstart**
δ, ε	**für Nachrichtenübertragungszeiten** t **gilt** $\delta - \varepsilon \le t' - t \le \delta + \varepsilon$
R	**Rundenabstand**

Uhrensynchronisation: LL

- **Lokale Variable**

ADJUSTMENT, AVERAGE

Hilfsgrößen bei der Berechnung der Zeitkorrekturen

ARRIVAL_TIMES[1 .. n]

Ankunftszeit der letzten Nachricht eines jeden Prozesses, gemessen in lokaler Zeit

CORRECTION

Zeitkorrektur

FLAG

BCAST: eigener Kenntnisstand ist zu verbreiten

UPDATE: eigener Kenntnisstand ist anzupassen

T

anfänglich T^0; nimmt der Reihe nach die Werte

$T^0 + R, T^0 + 2R, T^0 + 3R, \ldots$ **an (Anfangszeitpunkte der Runden).**

Methoden

reduce	entfernt aus einer Multimenge die f größten und die f kleinsten Werte
mid	berechnet den Mittelwert zwischen dem größten und dem kleinsten Element einer Multimenge

- Übergangsfunktion

receive(m) from q:

```
ARRIVAL_TIMES[q] = local_time();
```

// Start exchanging clock messages

(receive(START) or receive(TIMER)) and (FLAG == BCAST):

```
broadcast(T);
set_timer(T + (1 + ρ)(β + δ + ε));
FLAG = UPDATE;
```

// Compute new time correction

receive(TIMER) and (FLAG == UPDATE):

```
AVERAGE = mid(reduce(ARRIVAL_TIMES));
ADJUSTMENT = T + δ - AVERAGE;
CORRECTION += ADJUSTMENT;
T += R;
set_timer(T);
FLAG = BCAST;
```

S5.1 **Satz**

Der Algorithmus synchronisert wie erwünscht, wenn

$R > 2(1+\rho)(\beta+\varepsilon) + (1+\rho)\max\{\delta, \beta+\varepsilon\} + \rho\delta$ **und**

$R \le \beta/(4\rho) - \varepsilon/\rho - \rho(\beta+\delta+\varepsilon) - 2\beta - \delta - 2\varepsilon$,

woraus als hinreichende Bedingung folgt: $\beta \ge 4\varepsilon + 4\rho(3\beta+\delta+3\varepsilon) + 8\rho^2(\beta+\delta+\varepsilon)$,

also in erster Näherung $\beta \approx 4\varepsilon + 4\rho R$ **.**

Es sei c_p^i **die Uhr, die Prozessor p im Intervall** T^i **bis** T_p^{i+1} **benutzt.**

S5.2 **Satz**

Wenn p und q fehlerfreie Prozessoren sind, dann gilt

$$\left|c_p^i(T) - c_q^i(T)\right| \le (\beta + 2\rho(1+\rho)(\beta+\delta+\varepsilon))$$

für $T^0 - (1+\rho)(\beta+\delta+\varepsilon) \le T \le T^0 + (1+\rho)(\beta+\delta+\varepsilon)$ **wenn** $i = 0$ **ist und**

für $T^{i-1} - R - (1+\rho)(\beta+\delta+\varepsilon) \le T \le T^i + (1+\rho)(\beta+\delta+\varepsilon)$ **wenn** $i > 0$ **ist.**

S5.3 **Satz**

Für alle Zeitpunkte ist $|C_p(t) - C_q(t)| \le \gamma$

mit $\gamma = \beta + \varepsilon + \rho(7\beta + 3\delta + 7\varepsilon) + (8\rho^2 + 4\rho^3)(\beta+\delta+\varepsilon)$.

5.3.3 Internet-Synchronisationsprotokoll NTP (Network Time Protocol)

NTP und begleitende Informationen sind im WWW zu finden unter

http://sunsite.cnlab-switch.ch

Grundlegende Anforderungen

1. Primäre Zeitreferenzquellen müssen mit der amtlichen Zeit synchronisiert sein. Sie müssen auch Schaltsekunden der UTC berücksichtigen.
2. Die Zeitgeber müssen genaue Zeitangaben liefern, auch wenn die Übertragungszeiten stark schwanken.
3. Das für die Synchronisation genutzte Teilnetz muß verläßlich sein, auch wenn das Netzwerk instabil ist und Verbindungen für mehrere Tage ausfallen.
4. Das Synchronisationsprotokoll muß ständig arbeiten und den Zeitangleich so häufig vornehmen, daß erwartete Frequenzänderungen, z. B. als Folge von Temperaturschwanken, aufgefangen werden können.
5. Das System muß auf der Basis existierender Netze und für das gesamte Rechnerspektrum - vom Arbeitsplatzrechner bis zu Höchstleistungsrechner - funktionieren.

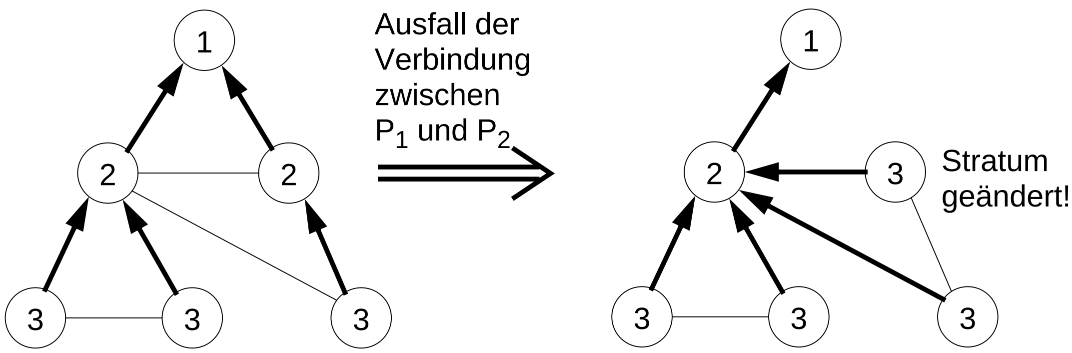
❑ Charakteristische Eigenschaften von NTP

1. Das Teilnetz für die Synchronisation ist ein selbstorganisierendes, hierarchisches Netzwerk von Zeitgebern.
2. Das Synchronisationsprotokoll arbeitet verbindungslos, um Verzögerungen zu minimieren, Implementierungen zu vereinfachen und überall nutzbar zu sein.
3. Das Synchronisationsprotokoll arbeitet symmetrisch. Es toleriert Paketverluste, Paketverdopplungen und Überholung.
4. Das Protokoll basiert auf der Idee der Phase-Locked-Loop.
5. Mehrfach redundante Zeigeber und Transportwege werden benutzt.
6. Der Zusatzaufwand wird mit Hilfe dynamischer Messungen auf ein Mindestmaß reduziert.

Uhrensynchronisation: Internet-Synchronisationsprotokoll NTP

- Klassen von Zeitgebern
  1. primär: Direkt (durch Hardware) mit amtlicher Zeit synchronisiert
  2. sekundär: Durch Datenaustausch mit primären Zeitgebern synchronisiert
- Kennzeichnung der Genauigkeitsklasse durch „Stratum“
  - 1: primärer Zeitgeber
  - i >1: synchronisiert mit Zeitgeber des Stratums i-1

Stratum kann dynamisch wechseln



Uhrensynchronisation: Internet-Synchronisationsprotokoll NTP

- Messungen
  - Meßgrößen

Zeitdifferenz	(clock offset)
Umlaufdauer	(roundtrip delay)
Streufehler	(dispersion)

  - Bezeichnungen

r	Ganggenauigkeit einer Uhr
f	maximaler Ablesefehler (bedingt durch Quantisierung)
Θ	Zeitdifferenz
δ	Umlaufdauer
ε	Streufehler
σ	Fehlerintervall

BP 1 Uhrensynchronisation: Internet-Synchronisationsprotokoll NTP

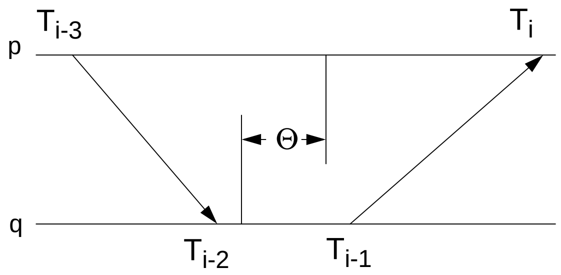
◆ Fehlerquellen

- Netzwerk

$$\Theta_i = \frac{T_{i-2} - T_{i-3} + T_{i-1} - T_i}{2}$$

$$\delta_i = T_{i-2} - T_{i-3} - T_{i-1} + T_i$$

$$\Theta_i - \frac{\delta_i}{2} \le \Theta \le \Theta_i + \frac{\delta_i}{2}$$



BP 1 Uhrensynchronisation: Internet-Synchronisationsprotokoll NTP

- Meßfehler

T_{i-3}	$[-f_p, 0]$	(Quantisierungsfehler)
T_{i-2}	$[-f_q, 0]$	(Quantisierungsfehler)
T_{i-1}	$[-f_p, 0] + \rho_q(T_{i-1} - T_{i-2})$	
T_i	$[-f_p, 0] + \rho_p(T_i - T_{i-3})$	

$$\sigma_{\delta_i} = [-f_p - f_q, f_p + f_q] + \rho_p(T_i - T_{i-3}) + \rho_q(T_{i-1} - T_{i-2})$$

$$\sigma_{\Theta_i} = [-f_q, f_p] + \frac{\rho_q(T_{i-1} - T_{i-2}) - \rho_p(T_i - T_{i-3})}{2}$$

Annahme: $\rho_p = \rho_q = \rho;\ f_p = f_q = f$

$$\sigma_{\delta_i} = [-2f, 2f] + \rho(T_i - T_{i-1} + T_{i-2} - T_{i-3})$$

$$\sigma_{\Theta_i} = \frac{[-f, f] + \rho(T_i - T_{i-3} + T_{i-2} - T_{i-1})}{2}$$

Uhrensynchronisation: Internet-Synchronisationsprotokoll NTP

$$\varepsilon_{\delta_i} = 2f + \rho(T_i - T_{i-1} + T_{i-2} - T_{i-3})$$

$$\varepsilon_{\Theta_i} = \frac{f + \rho(T_i - T_{i-3} + T_{i-2} - T_{i-1})}{2}$$

Zusammengefaßt

$$\Theta_i - \varepsilon_{\Theta_i} - \frac{\delta_i + \varepsilon_{\delta_i}}{2} \le \Theta \le \Theta_i + \varepsilon_{\Theta_i} + \frac{\delta_i + \varepsilon_{\delta_i}}{2}$$

peer dispersion: $\varepsilon = \varepsilon_{\Theta_i} + \frac{\delta_i + \varepsilon_{\delta_i}}{2}$

Korrektheitsintervall: $I = \left[\Theta - \frac{\delta}{2} - \varepsilon, \Theta + \frac{\delta}{2} + \varepsilon\right]$

Uhrensynchronisation: Internet-Synchronisationsprotokoll NTP

- Auswahl der Referenzuhr:
  1. Auswahl nach einem Genauigkeitsindex

     Für jeden in Betracht gezogenen Zeitgeber wird seine Genauigkeit ermittelt:

     Mehrere Paare (Θ_i, δ_i) mit $0 \le i \le n$ werden nach wachsendem δ geordnet.
     Paar mit kleinstem δ wird als Schätzung genommen.

     Streuung als Qualitätsmaß: $\varepsilon = \sum_{j=0}^{n-1} |\Theta_i - \Theta_0| v^j$ mit $v \le 1$, typisch $v = \frac{1}{2}$

     Ausgewählt werden Zeitgeber mit kleinem ε.
  2. Es werden hinreichend viele Zeitgeber mit überschneidenden Korrektheitsintervallen ausgewählt.
  3. Die im Schritt 2 ausgewählten Zeitgeber werden erstens nach Stratum und zweitens nach steigender Umlaufdauer zwischen ihm und seinem primären Zeitgeber geordnet. Davon werden die ersten m (fünf) betrachtet.
     - Für jeden wird die Streuung $q_i = \sum_{k=0}^{m-1} |\Theta_i - \Theta_k| w^k$ berechnet mit $w \le 1$.
     - Derjenige mit maximalem q_i wird aus der Betrachtung ausgeschlossen.
     - Die beiden vorangehenden Schritte werden wiederholt, bis nur ein Zeitgeber übrig ist.
     - Nach dessen Uhr wird die lokale Uhr gestellt.

Uhrensynchronisation: Internet-Synchronisationsprotokoll NTP