

## 3.5 Das Network Time Protocol - NTP

- NTP und begleitende Informationen sind im WWW zu finden unter  
`http://www.ntp.org` ("Offizielle" NTP-Homepage)  
sowie  
`http://www.eecis.udel.edu/~mills` (Homepage David Mills)
  
- Geschichte
  - ◆ Entwickelt seit 1982 (NTP v1, RFC 1059) unter Leitung von David Mills
  - ◆ Seit 1990 NTP v3, teilweise immer noch in Verwendung
  - ◆ Aktuelle Version NTP v4, seit 1994
  - ◆ Aktuelle Forschung von Mills (University of Delaware): Interplanetare Zeitsynchronisation für IPIN (Interplanetary Internet, NASA/DARPA-Projekt)

## 3.5 Das Network Time Protocol - NTP

- Eigenschaften von NTP
  - ◆ Zweck: Synchronisierung von Rechneruhren im bestehenden Internet
  - ◆ Derzeit weit über 100.000 NTP-Knoten weltweit
    - 107 aktive öffentliche Stratum 1 - Knoten (Stand Mai 2001)
    - 136 aktive öffentliche Stratum 2 - Knoten
  - ◆ Erreichbare Genauigkeiten von ca 0.01s in WANs, < 1ms in LANs
  - ◆ NTP-Dämon auf fast allen Rechnerplattformen verfügbar, von PCs bis Crays; Unix, Windows, VMS, eingebettete Systeme
  - ◆ Fehlertolerant

## 3.5 Das Network Time Protocol - NTP

### ■ Grundlegener Überblick

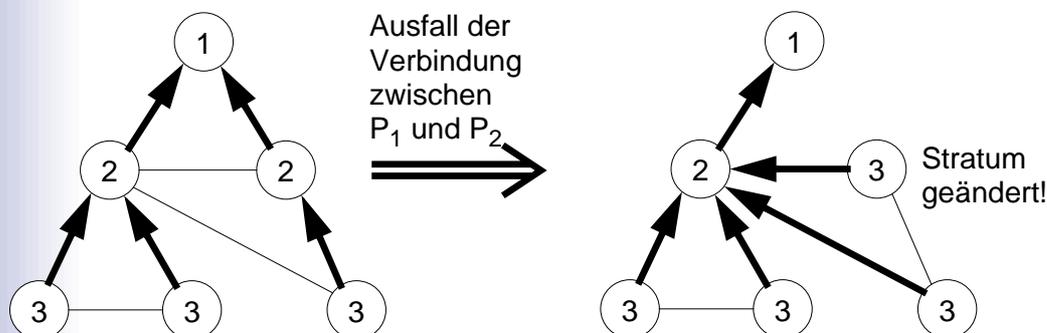
- ◆ Primäre Server (Stratum 1), über Funk oder Standleitungen an amtliche Zeitstandards angebunden
- ◆ Sekundäre Server und Clients synchronisieren sich nach primären Servern über selbstorganisierendes, hierarchisches Netz
- ◆ Verschiedene Betriebsarten (Master/Slave, symmetrische Synchronisation, Multicast, jeweils mit/ohne kryptographischer Authentisierung)
- ◆ Zuverlässigkeit durch redundante Server und Netzwerkpfade
- ◆ Optimierte Algorithmen, um Fehler durch Jitter, wechselnde Referenzuhren und fehlerhafte Server zu reduzieren
- ◆ Lokale Uhren werden in Zeit und Frequenz durch einen adaptiven Algorithmus geregelt.

## 3.5 Das Network Time Protocol - NTP

### ■ Stratum:

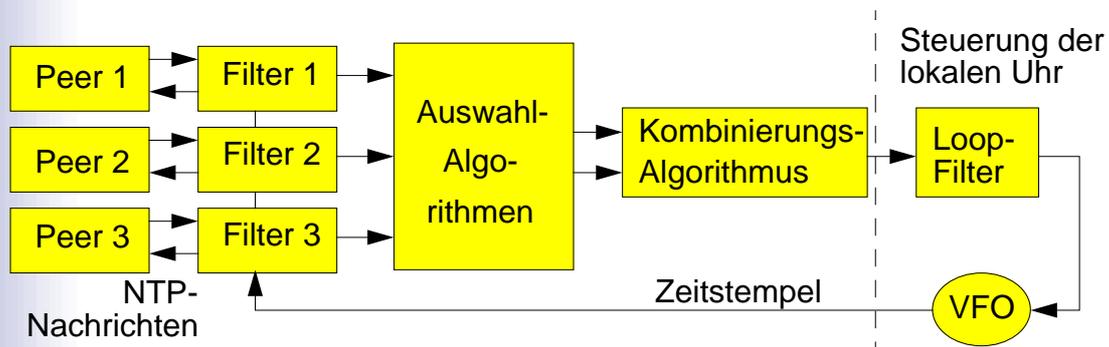
- ◆ 1: primärer Zeitgeber
- ◆  $i > 1$ : synchronisiert mit Zeitgeber des Stratums  $i-1$

### ■ Stratum kann dynamisch wechseln:



## 3.5 Das Network Time Protocol - NTP

### ■ Architektur-Überblick



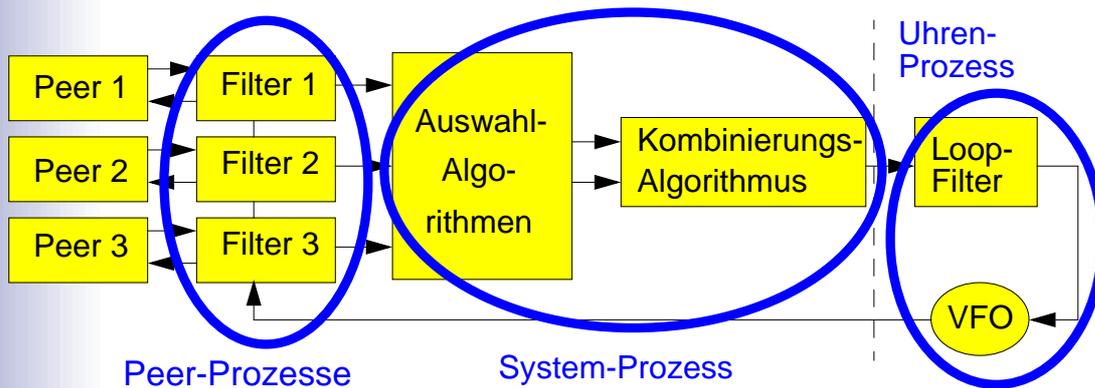
## 3.5 Das Network Time Protocol - NTP

### ■ Architektur-Überblick (Fortsetzung...)

- ◆ Mehrere Referenzserver ("Peer") für Redundanz und Fehlerstreuung
- ◆ Peer-Filter wählen pro Referenzserver den jeweils besten Wert aus den acht letzten Offset-Messwerten aus
- ◆ Die Auswahlalgorithmen versuchen zunächst richtiggehende Uhren ("truechimers") zu erkennen und falsche Uhren ("falsetickers") auszufiltern, und wählen dann möglichst genaue Referenzuhren aus
- ◆ Der Kombinationsalgorithmus berechnet einen gewichteten Mittelwert der Offset-Werte (frühere NTP-Versionen wählen einfach den am vertrauenswürdigsten erscheinenden Referenzknoten aus)
- ◆ Loop Filter und VFO bilden zusammen die geregelte lokale Uhr. Die Regelung ist so entworfen, dass Jitter und Drift bei der Regelung minimiert werden

## 3.5 Das Network Time Protocol - NTP

### ■ NTP-Prozesse

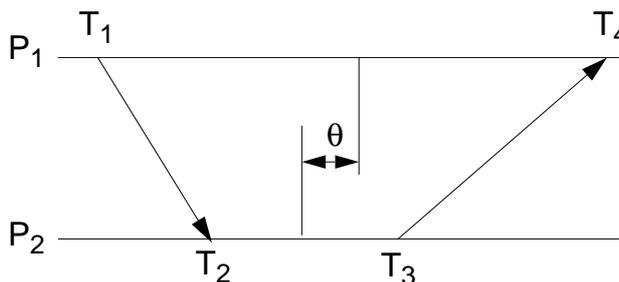


- ◆ Peer-Prozesse: Unabhängig in periodischen Abständen; Periode durch dynamisch System-Prozess und durch entfernte Server bestimmt
- ◆ System-Prozess: Periodisch, Abstände dynamisch aus gemessenem Phasen-Jitter und Stabilität der lokalen Uhr bestimmt
- ◆ Uhren-Prozess: Periodisch in 1s-Intervallen (Regelung VFO-Phase und -Frequenz)

## 3.5 Das Network Time Protocol - NTP

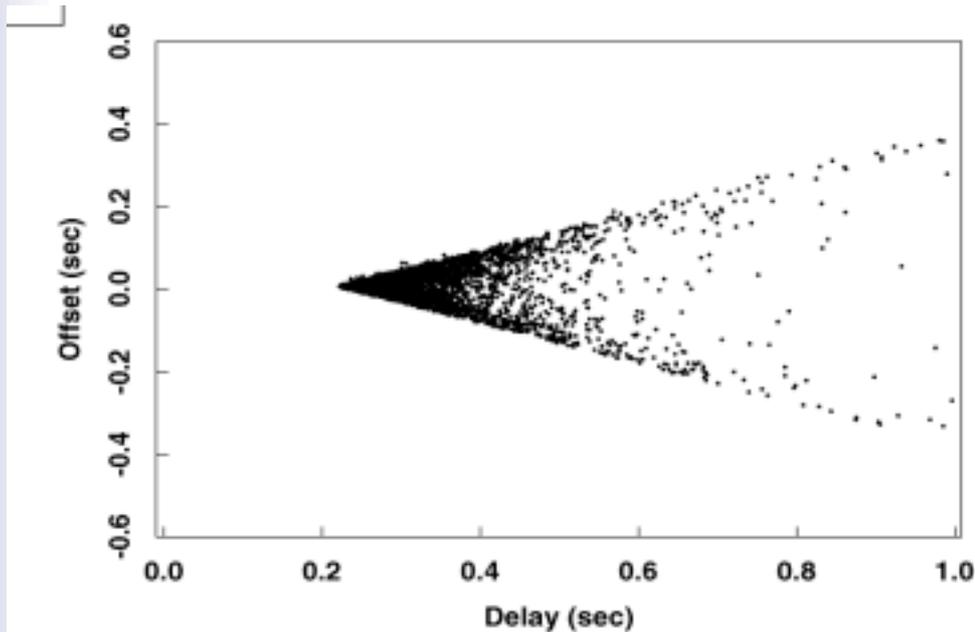
### ■ Offset-Messung

- ◆ Es werden die letzten 8 Messungen gespeichert; Index  $i=0$ : neueste Messung



- ◆ Reine Nachrichtenlaufzeit:  $\delta = (T_4 - T_1) - (T_3 - T_2)$
- ◆ Geschätztes Offset:  $\theta = (T_2 + T_3)/2 - (T_1 + T_4)/2$ 
  - Exakter Wert, falls Laufzeiten in beide Richtungen gleich sind
  - Maximaler Fehler bei unsymmetrischen Laufzeiten:  $\delta/2$

## 3.5 Das Network Time Protocol - NTP



- ◆ Bei kleinstem Delay  $\delta$  ist der gemessener Offsetwert  $\theta$  am genauesten!

## 3.5 Das Network Time Protocol - NTP

### Aufbau der NTP Datenpakete

LI	VN	Mode	Strat	Poll	Prec
Root Delay					
Root Dispersion					
Reference Identifier					
Reference Timestamp (64)					
Originate Timestamp (64)					
Receive Timestamp (64)					
Transmit Timestamp (64)					
Extension Field 1 (optional)					
Extension Field 2 (optional)					
Key/Algorithm Identifier					
Message Hash (64 or 128)					

LI leap indicator  
 VN version number  
 Strat stratum (0..15)  
 Poll poll interval (log2)  
 Prec precision (log2)

### NTP Zeitstempel-Format

Seconds (32) | Fraction (32)

Sekunden seit dem 1.1.1900

Extension Field: nur NTPv4

## 3.5 Das Network Time Protocol - NTP

- Der Peer-Filter-Algorithmus: Ausgangsdaten
  - ◆ Bei jeder Messung ermittelte Werte:
    - Offset  $\theta$ , Laufzeit  $\delta$
    - Abschätzung des Messfehlers:  
 $\varepsilon = \text{Lesegenauigkeit} + \text{MAXDRIFT} * (T_4 - T_1)$ ,
  - ◆ Eingetragene Messwerte werden in einen Puffer eingetragen
    - Speicherung der letzten 8 Messwerte
    - Aktualisierung der Fehlerabschätzung  $\varepsilon$  bei jeder neuen Messung um den möglichen Fehler durch Alterung (Uhrendrift)
      - $i=7..1$ :  $(\delta_i, \theta_i, \varepsilon_i) = (\delta_{i-1}, \theta_{i-1}, \varepsilon_{i-1} + \text{MAXDRIFT} * \text{verstrichene Zeit})$
      - $i=0$ :  $(\delta_0, \theta_0, \varepsilon_0) = \text{neuer Messwert}$

## 3.5 Das Network Time Protocol - NTP

- Der Peer-Filter-Algorithmus: Filterung
  - ◆ Eingangsdaten: Liste von Messwerten  $(\delta_i, \theta_i, \varepsilon_i)$
  - ◆ Sortierung der Messwerte nach  $\varepsilon_i + \delta_i/2$ ;  
Minimaler Wert wird als Referenz verwendet
  - ◆ Berechnung der Filter-Streuung  $\varepsilon_\sigma$  als gewichtete Abweichung des Messwerts relativ zu den anderen, sortierten Messwerten (als Mass für die Genauigkeit der Messwerte)

$$\varepsilon_\sigma = \sum_{i=0}^7 |\theta_i - \theta_0| v^i \quad \text{mit } v=0.5 \quad (\text{experimentell})$$

- ◆ Das Ergebnis ist für jedes Peer ein Tripel aus Delay, Offset, Dispersion:  
 $(\delta, \theta, \varepsilon) = (\delta_0, \theta_0, \varepsilon_0 + \varepsilon_\sigma)$
- ◆ NTP v4: Zusätzlich Berechnung der Varianz der Messwerte

## 3.5 Das Network Time Protocol - NTP

### ■ Variablen-Berechnung für Systemprozess

- ◆ Für jedes Peer werden folgende Variablen berechnet:

$$\Theta = \theta$$

$$\Delta = \text{RootDelay} + \delta$$

$$E = \text{RootDispersion} + \varepsilon + (\text{Alter der Messung}) * \text{MAXDRIFT}$$

⇒ Tripel  $(\Theta_i, \Delta_i, E_i)$  für das Peer  $i$

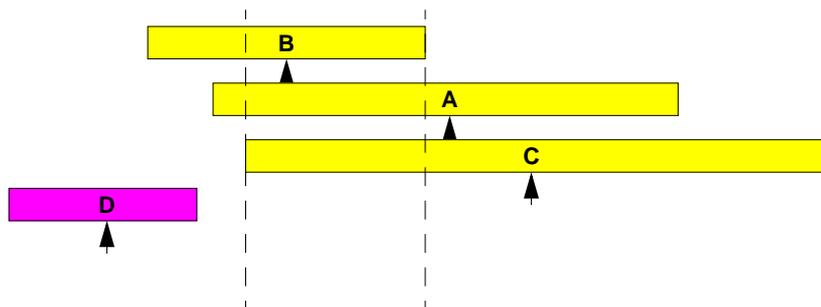
- ◆ Weitergabe an den Systemprozess, dieser berechnet daraus:

$$\text{Synchronisierungsdistanz: } \Lambda_i = E_i + \Delta_i/2$$

$$\text{Korrektheitsintervall: } [\Theta_i - \Lambda_i, \Theta_i + \Lambda_i]$$

## 3.5 Das Network Time Protocol - NTP

### ■ Auswahl-Algorithmus



- ◆ Trennung von "truechimers" und "falsetickers"
- ◆ DTS (Digital Time Service, Vorläufer-Algorithmus, einfach):  
Ermittle Durchschnitt mit den meisten überlappenden Korrektheitsintervallen  
Mittelpunkt des Intervalls wird als Offset zur Uhrenkorrektur verwendet
- ◆ Ziel bei NTP: Auswahl des Intervalls so, dass die Mittelpunkte der Intervalle der als korrekt betrachteten Knoten im Intervall liegen

## 3.5 Das Network Time Protocol - NTP

### ■ Clustering-Algorithmus

- ◆ Ziel: Weitere Selektion von möglichst "guten" Referenzen dabei: kleines Stratum und kleine Synchronisationsdistanz bevorzugen
- ◆ Sortierung der Liste der Referenzen nach der Metrik  
"Stratum \* MAXDISPERSE + Synchronisationsdistanz"
- ◆ Für jedes Peer k wird eine gewichtete Streuung  $\varepsilon_{\xi k}$  relativ zu den anderen Peers berechnet

$$\varepsilon_{\xi k} = \sum_{i=0}^7 |\theta_i - \theta_k| w^i \quad \text{mit } w = 0.75$$

- ◆ Der Knoten mit maximalen  $\varepsilon_{\xi k}$  wird entfernt, und das ganze von vorne wiederholt, solange
  - es in der Liste mehr als MINCLOCK Knoten gibt
  - das maximale  $\varepsilon_{\xi k}$  grösser als das minimale  $E_i$  ist

## 3.5 Das Network Time Protocol - NTP

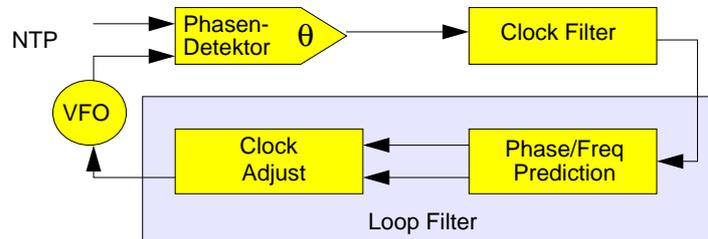
### ■ Kombinationsalgorithmus

- ◆ Die übrigbleibenden Knoten werden zur Synchronisation verwendet
  - Einfache Variante: Falls der bisherige Referenzknoten sich in der Liste verwendet, wird dieser weiterhin zur Synchronisation verwendet; ansonsten wird der Knoten am Anfang der Liste zum neuen Synchronisationsknoten
  - Komplexe Variante (NTPv4): Berechnung eines gewichteten Mittelwerts der Offsets aus allen Knoten
- ◆ Das so ermittelte Offset wird an die Uhrenregelung (Clock disciplin algorithm) weitergegeben

## 3.5 Das Network Time Protocol - NTP

### ■ Uhrenregelung

- ◆ Regelungsprinzip: Hybride PLL/FLL-Regelschleife
- ◆ Regelungstechnische Modellierung:



- ◆ Ziel der hybriden Regelung:

- Frequenzregelung zum Ausgleich von Frequenzfehlern und langfristigen Schwankungen
- Phasenregelung zur Korrektur von kurzfristigen Schwankungen

## 3.5 Das Network Time Protocol - NTP

### ■ Praktische Beispiele

```
$ ntpq
```

```
ntpq> host faui02
current host set to faui02.informatik.uni-erlangen.de
```

```
ntpq> peers
=====
remote          refid          st t when poll reach  delay  offset  disp
=====
LOCAL(1)        LOCAL(1)      6 l  26  64  377   0.00   0.000  10.06
*ntp0-rz.rrze.un .GPS.         1 u  30 1024 377   1.68   0.097   0.31
+ntp1-rz.rrze.un .DCFp.        1 u 111 1024 377   1.82   0.189   0.15
+ntp2-rz.rrze.un ntp1-rz.rrze.un 2 u  51 1024 377   1.54  -0.064   0.35
ntp3-rz.rrze.un 0.0.0.0      16 u 430 1024  0    0.00   0.000 16000.0
```

```
ntpq> rv
status=0664 leap_none, sync_ntp, 6 events, event_peer/strat_chg
system="SunOS", leap=00, stratum=2, rootdelay=1.68,
rootdispersion=10.97, peer=46269, refid=ntp0-rz.rrze.uni-erlangen.de,
reftime=c1611784.b5b01000 Wed, Oct 23 2002 14:41:40.709, poll=10,
clock=c161183b.c68fc000 Wed, Oct 23 2002 14:44:43.775, phase=0.143,
freq=3584.06, error=0.49
```

## 3.6 Zusammenfassung

---

- Zeit in verteilten Systemen ist ein zentrales Problem
- Wenn Aussagen zur kausalen Ordnung von Ereignissen benötigt werden, bietet sich der Einsatz von logischen Uhren an
  - ◆ Logische Uhren nach Lamport, erweiterbar zu einer totalen Ordnung auf alle Ereignisse, die kausale Beziehungen respektiert
  - ◆ Vektoruhren zur exakten Erfassung von kausalen Beziehungen
- Das Network Time Protocol (NTP) bietet die Möglichkeit, lokale Uhren mit für viele Zwecke ausreichender Genauigkeit an die offizielle Zeit zu synchronisieren
  - ◆ Hierarchisches, fehlertolerantes Synchronisationsnetzwerk