

# Verteilte Systeme – Übung

## Zeit in verteilten Systemen

---

Sommersemester 2024

Harald Böhm, Laura Lawniczak, Tobias Distler

Friedrich-Alexander-Universität Erlangen-Nürnberg  
Lehrstuhl für Informatik 4 (Systemsoftware)

<https://sys.cs.fau.de>



**Lehrstuhl für Informatik 4**  
Systemsoftware



**Friedrich-Alexander-Universität**  
Technische Fakultät

Zeit in verteilten Systemen

Echtzeit-basierte Uhren

Logische Uhren

- Ist Ereignis A auf Knoten X passiert, bevor Ereignis B auf Knoten Y passiert ist?

Beispiele: Internet-Auktionen, Industriesteuerungen, ...

- Prinzipiell keine konsistente Sicht auf Gesamtsystem möglich
  - Unabhängigkeit von Ereignissen
  - Informationsaustausch mit Latenzen verbunden

⇒ Nur näherungsweise Lösungen möglich

- Bestes Verfahren abhängig von Einsatzgebiet und notwendigen Eigenschaften

## **Zeit in verteilten Systemen**

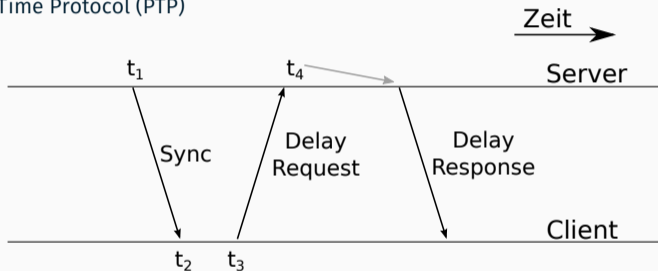
---

### **Echtzeit-basierte Uhren**

- Nutzung eines gemeinsamen Zeitsignals
  - Auflösung beschränkt
  - Schwierig über größere Entfernungen
    - Ausbreitungsgeschwindigkeit: max. 30 cm/ns
- Nachrichten mit Zeitstempel lokaler, physikalischer Uhren versehen
  - Wenig Kommunikationsaufwand
  - Ohne Synchronisation: Zunehmende Abweichungen
- Kombination verschiedener Verfahren zur Verbesserung der Genauigkeit

# Synchronisation von Echtzeituhren: NTP, PTP

- Stellen lokaler Uhr basierend auf Referenzuhr
- In der Praxis verwendete Protokolle:
  - Network Time Protocol (NTP)
  - Precision Time Protocol (PTP)



- Berechnung von Umlaufzeit & Verzögerung anhand von Zeitstempel
- Annahmen: Laufzeiten symmetrisch und stabil
- Genauigkeit über Internet in der Größenordnung 10 ms

- Messung von Neutrino-Flugzeit zwischen CERN und LNGS (732 km)
- Möglichst genaue Zeitsynchronisation zwischen Standorten
- White Rabbit: Kombination verschiedener Techniken
  - Synchronous Ethernet über Glasfaser
  - Atomuhren als Taktgeber
  - Precision Time Protocol (PTP) mit Hardware-Unterstützung
  - Global Positioning System (GPS)
- Ausgleich von Temperaturschwankungen durch ständige Phasen-Messung
- Genauigkeit: 0.5 ns, Präzision: 10 ps (5 km Teststrecke)



M. Lipiński, T. Włostowski, J. Serrano, and P. Alvarez.

**White Rabbit: a PTP Application for Robust Sub-nanosecond Synchronization.**

*2011 International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS'11)*, p. 25–30, September 2011.

# Zeit in verteilten Systemen

---

## Logische Uhren



## Grundidee

Kausale Zusammenhänge entstehen durch gegenseitige Beeinflussung, d. h. Nachrichtenaustausch in verteiltem System

## Modell

Kommunizierende Prozesse  $P_i$  versehen Ereignisse  $a$  mit logischem Zeitstempel  $C_i\langle a \rangle$

## Uhrenbedingung

Wenn Ereignis  $b$  aufgrund von  $a$  aufgetreten ist ( $a \rightarrow b$ ), muss die Relation  $C_i\langle a \rangle < C_j\langle b \rangle$  gelten

- Eigenschaften: transitiv, asymmetrisch  $\Rightarrow$  Striktordnung
  - $\rightarrow$  Umkehrschluss **nicht** möglich: Aus  $C_i\langle a \rangle < C_j\langle b \rangle$  folgt nicht  $a \rightarrow b$ !
- Erweiterte Ansätze können zusätzliche Eigenschaften garantieren
  - Totalordnung
  - Zuverlässige Unterscheidung abhängiger Ereignisse ( $\rightarrow$  Vektoruhr)

- Uhrenbedingung im Kontext von kommunizierenden Prozessen
  1. Aufeinanderfolgende Ereignisse innerhalb eines Prozesses erhalten streng monoton steigende Zeitstempel
  2. Senden einer Nachricht muss vor deren Empfang passiert sein, daher muss gelten:

$$C_i\langle\text{Senden}\rangle < C_j\langle\text{Empfang}\rangle$$

- Regeln für **Implementierung**

1. Die logische Uhr  $C_i$  eines Prozesses  $P_i$  muss zwischen zwei aufeinanderfolgenden Ereignissen immer inkrementiert werden
2. Erhält ein Prozess  $P_j$  eine Nachricht und deren Zeitstempel  $C_i\langle\text{Senden}\rangle$  ist größer oder gleich dem Wert der Uhr  $C_j$  des Prozesses  $P_j$ , muss die Uhr auf einen Wert größer  $C_i\langle\text{Senden}\rangle$  erhöht werden



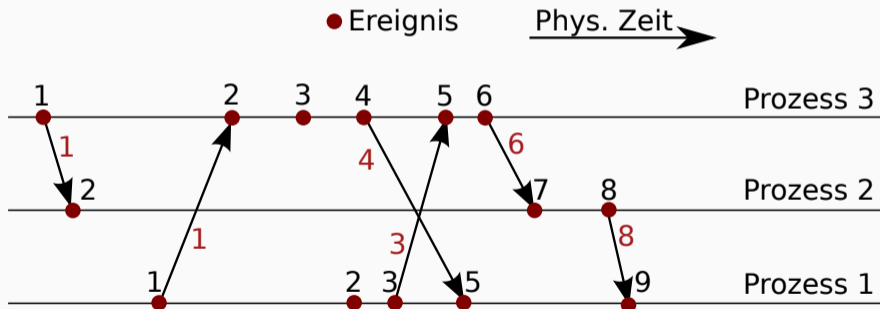
Leslie Lamport.

**Time, Clocks, and the Ordering of Events in a Distributed System.**

*Communications of the ACM*, 21:558–565, July 1978.

# Uhrenbedingung von Lamport

- Kein genereller Zusammenhang mit Ablauf physikalischer Zeit
  - Kein gleichmäßiger Verlauf
  - Folge von Ereignissen nach logischer Zeit nicht zwangsläufig identisch mit physikalischem Auftreten



- Für viele Anwendungen Totalordnung wünschenswert
  - Wenn Zeitstempel  $C_i\langle a \rangle$  und  $C_j\langle b \rangle$  gleich, gilt weder  $C_i\langle a \rangle < C_j\langle b \rangle$ , noch  $C_j\langle b \rangle < C_i\langle a \rangle$
  - Beliebiges **determiniertes** Verfahren zur Festlegung möglich
  - Am einfachsten: Global eindeutige Prozess-ID entscheidet
  - Keine Beeinflussung der Aussage bezüglich kausaler Zusammenhänge
  
- Implementierung von Relationen in Java mittels Comparable

```
public interface Comparable<T> {  
    public int compareTo(T obj);  
}
```

- Methode `compareTo()` liefert Zahl abhängig von Relation

**Negativ** : `this < obj`

**„Null“** : `this = obj`, entspricht `equals()`

**Positiv** : `this > obj`