HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

S1
Zeit in verteilten Systemen

ZEHN DINGE die Sie noch nicht wissen über

Zeit

- Die Rotation der Erde verlangsamt sich stetig. Die Länge eines Tages nimmt dadurch pro Jahrhundert um etwa drei Millisekunden zu.

- Im Rückblick verkehrt sich die Zeitempfindung: In Zeiträumen, in denen sehr viel geschehen ist und die einst wie im Flug vergangen sind, speichern Menschen sehr viel mehr Informationen. Dadurch erscheinen sie in der Erinnerung länger als ereignisarme, langweilige Perioden.

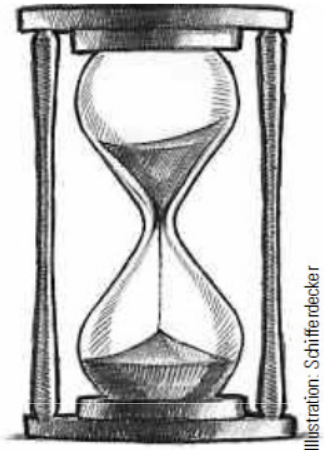


Illustration: Schifferdecker

- Die regelmäßigen Wechsel von Sommer- und Winterzeit verdankt die Welt Benjamin Franklin. In einem Brief an eine Pariser Zeitschrift schlug er vor, die Uhren im Sommer eine Stunde vorzustellen, um mehr Tageslicht zu nutzen und so den Verbrauch teurer Kerzen zu reduzieren. Ob das ernst gemeint oder nur ein Scherz war, ist bis heute unklar.

- Wer sich mit der Geschichte und Lehre der Sonnenuhren beschäftigt, betreibt Gnomonik. Der Begriff geht auf das griechische Wort Gnomon zurück, das Schattenstab bedeutet.

- Das Erdbeben in Chile am 27. Februar 2010 hat die Dauer des Tages um 1,26 Mikrosekunden verkürzt.

- Erst beim Ausbau der Eisenbahnnetze führten die Staaten standardisierte Zeiten ein. Zuvor hatten etwa in Großbritannien alle Gemeinden ihre lokale Zeit mittels Sonnenuhr definiert. Wegen der kleinen Abweichungen – jedes Dorf lag quasi in seiner eigenen Zeitzone – war es schwer Abfahrts- und Ankunftszeiten von Zügen zu bestimmen.

- Der schwedische Wissenschaftler Carl von Linné legte 1745 im botanischen Garten von Uppsala eine Blumenuhr an. Er hatte beobachtet, dass verschiedene Blumen ihre Blüten zu unterschiedlichen Tageszeiten öffnen.

- 0,000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 05 Sekunden: Das ist der kürzeste Zeitraum, für den die bekannten Gesetze der Physik anwendbar sind. Er wird Planck-Zeit genannt.

- Frauen brauchen angeblich länger als Männer, um einen Witz zu verstehen. Diese Erkenntnis ist Wissenschaftlern der Universität Stanford zu verdanken, die Probanden in Hirnscanner legten, sie Cartoons ansehen ließen und schließlich Hirnaktivitäten sowie die Zeit maßen, die die Probanden brauchten, um einen Scherz zu begreifen.

- Ein Tag hat 24 Stunden – auch das sollte zur Zeit der Französischen Revolution verändert werden. Der Vorschlag scheiterte jedoch, den Tag in zehn Stunden zu je 100 Minuten zu je 100 Sekunden zu unterteilen. Jede dieser „Sekunden" wäre dann um 14 Prozent kürzer gewesen als die bekannte Zeiteinheit. SEBASTIAN HERRMANN

Süddeutsche Zeitung
vom 1.10.2010

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Eigenschaften verteilter Systeme

- **Szenarien:**
  - ➔ konkurrierender Zugriff auf einmal vorhandene Betriebsmittel
  - ➔ verteilter Datenbankzugriff, ...

- **Ereignisabfolgen** könnten durch „exakte global gültige Zeitstempel“ gekennzeichnet werden:
  ➔ damit sind „vor“, „nach“ und „gleichzeitig“ Aussagen für Synchronisationsaufgaben möglich.

- **Aber:**
  - ➔ Zentrale Instanzen in einem verteilten System existieren i. d. R. nicht.
  - ➔ Es gibt keine gemeinsame Hardwareuhr die allen zur Verfügung steht.

- **Gegeben:** ➔ jeder beteiligte Rechner besitzt i. d. R. nur eine interne Rechneruhr.
- **Aufgabe:**
  - Synchronisierung der Uhren auf eine externe Referenzzeit: **externe Synchronisation**
  - Abgleich der Uhren der beteiligten Rechner untereinander: **interne Synchronisation**

„Rechnerexterne“ Zeitnormale (1)

Historisch: Die Dauer einer **Sekunde** (Basiswert) ist **astronomisch** 1/86400 eines **Solartages**.

- Die Erdrotation verlangsamt sich jedoch stetig.
- Zeitangabe auf astronomischer Basis ungenau.

Heute Atomuhr (erfunden 1948):

Die **Sekunde** ist die Dauer von 9 192 631 770 Übergangsperioden in Cäsium-133 Atomen

- Zeitdauer ist festgelegt auf Grund **physikalischer** Eigenschaften

- Die **aktuelle Atomuhr-Zeit in Sekunden** ist die **Anzahl** der Übergangsperioden dieser Cäsium-Uhr seit Mitternacht **1. Januar 1958** geteilt durch 9 192 631 770.
- Es gibt ca. 260 Atomuhren weltweit, die physikalisch-technische Bundesanstalt (PTB) betreibt mehrere in Deutschland (erfüllt gesetzlichen Auftrag „Zeitgesetz 1978“!!)
- Uhrendriftrate**:** pro Jahr ca. eine Abweichung von einer Millionstel Sekunde = $1{,}5 \times 10^{-14}$

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

„Rechnerexterne“ Zeitnormale (2)

- **Die International Atomic Time** (**TAI**)
  Mittelwert aus ca. 50 Atomzeiten nationaler Institute ausgerüstet mit ca. 250 Atomuhren weltweit in Paris im BIPM „bureau international des poids et mesures“.
  ➔ **Aber so unbrauchbar**!
  Ein Tag nach der **TAI** ist drei Millisekunden kürzer als ein **Solartag**

- **Aus TAI wird ab** 1967: **Universal Coordinated Time (UTC)** vom **BIPM** als gültige Referenzzeit veröffentlicht.
  - Unterschied zu TAI:
    BIPM fügt bei UTC eine Schaltsekunde ein, wenn die TAI und die Solarzeit um mindestens 800 Millisekunden auseinander liegen (Bisher ca. 30 Schaltsekunden)

- **Beispiele der „Veröffentlichungswege“ der UTC (über nationale Zeitinstitute) weltweit:**
  - UTC wird mittels GEOS (Geostationary Environmental Operational Satellites) ausgestrahlt,
  - UTC über das GPS (Global Positioning System) mit einer Genauigkeit von ca. 150 ns ,
  - UTC (PTB) über DCF77 Langwellensender bei Mainflingen,
  - **UTC (PTB) übers Internet via NTP-Protokoll, die erreichbare Sync-Genauigkeit beträgt 1-50 ms**

Uhrensynchronisation in verteilten Systemen (1)

- **Übers Internet: Network Time Protocol (NTP RFC-1305)**

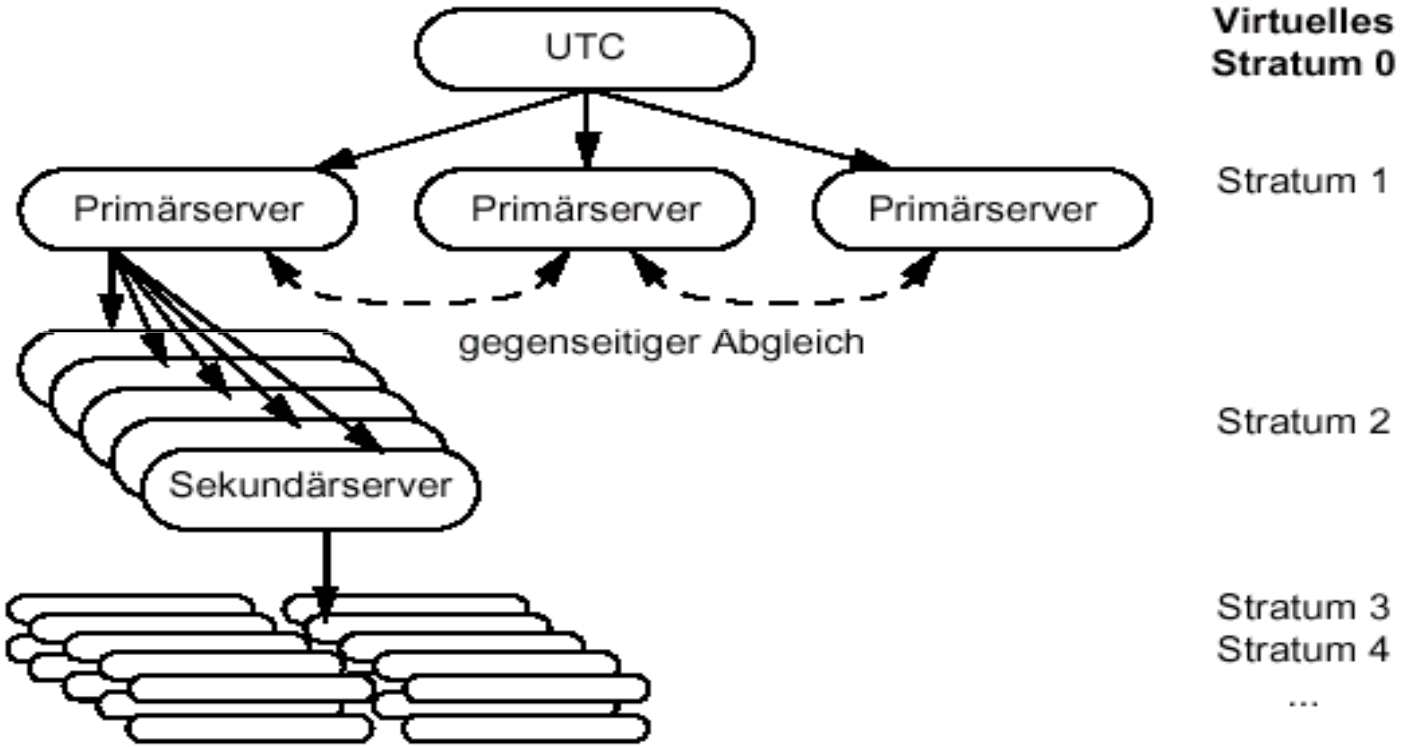


Bild Z.1: Baumhierarchie des NTP

Uhrensynchronisation in verteilten Systemen (2)

- **<u>verwendete Synchronisationsmodi bei NTP:</u>**

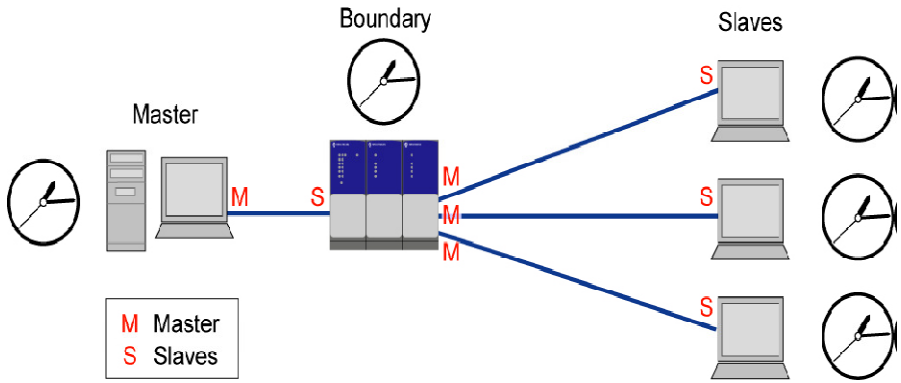
  - ***Multicast Modus* (➔UDP) *verwendet von Timeservern im LAN***
    - Im LAN wird die Zeit des Timeservers an alle Timeclients propagiert und es wird eine feste Verzögerung vom Timeserver zum Timeclient angenommen
      ➔ schlechte Genauigkeit

  - ***SNTP (Simple Network Time Protocol) verwendet von Timeclients***
    - **Timeclients** fragen via SNTP **Timeserver**, einfache Berücksichtigung von Roundtrip-/Offset-Delays

  - ***Symmetrischer Modus* (➔UDP) *verwendet zwischen Timeservern***
    - **Timeserver** des gleichen oder eines niedrigeren **Stratum** gleichen sich untereinander ab, indem sie sich wechselseitig NTP-Zeitinformationen zuschicken. Das Verfahren erreicht eine hohe Genauigkeit (ca. 30 ms, Synchronisationsdauer **zwischen 1-8 Minuten**).

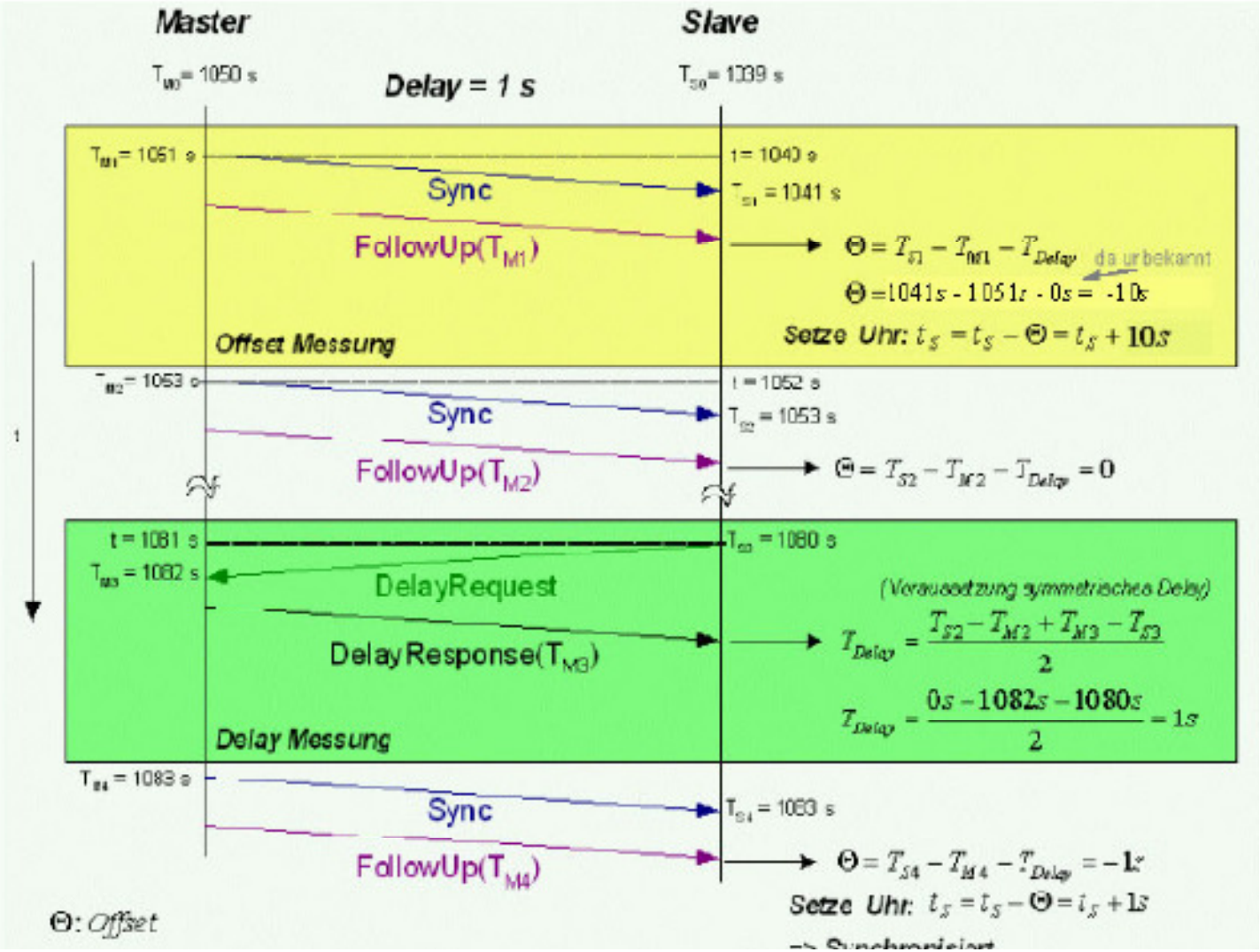
Uhrensynchronisation in verteilten Systemen (3)

- **Innerhalb lokaler ethernetbasierter Netze IEEE 1588 (Quelle Hirschmann):**



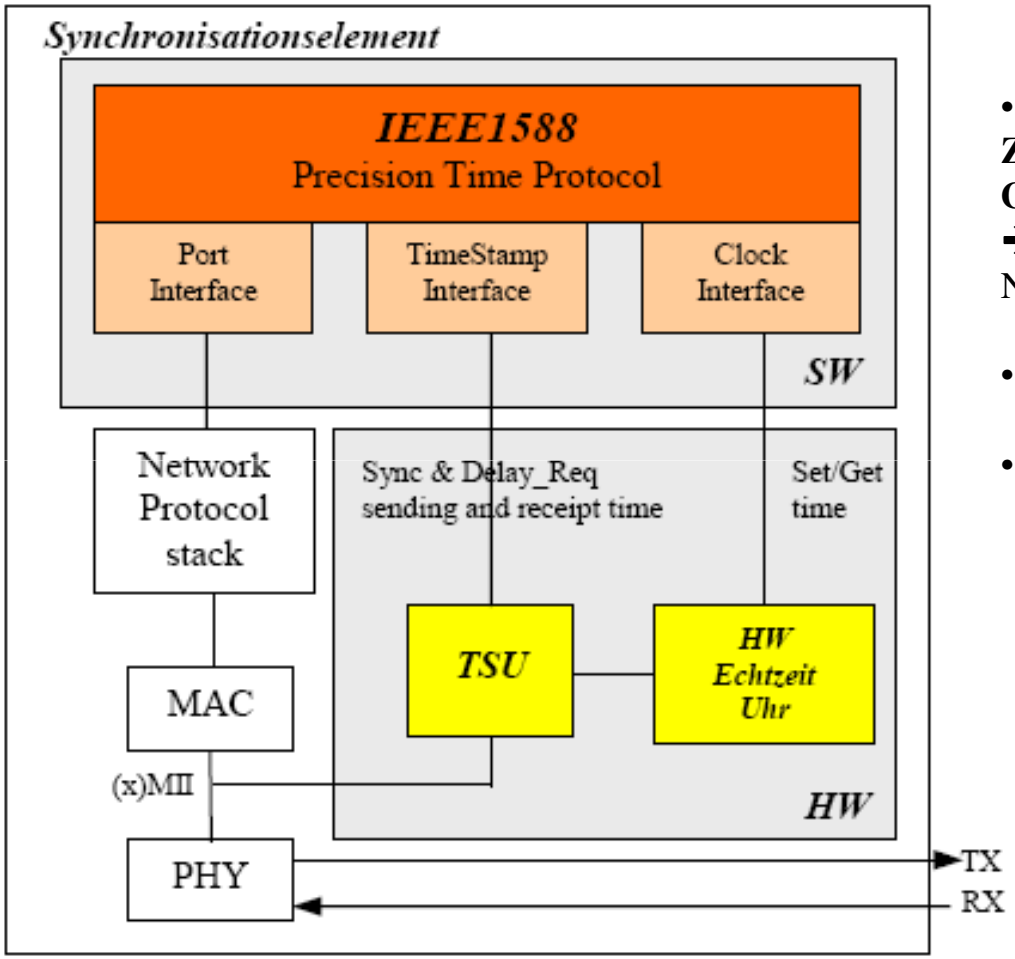
• Präzise Zeitinformation
➔ für verteilte Systeme in der Automatisierungstechnik wichtig.
• IEEE 1588 definiert ein **Precision Time Protocol** (PTP)
um in Ethernet-Netzwerken verteilte Uhren auf
weniger als eine Mikrosekunde Genauigkeit zu synchronisieren
PTP kennt zwei Arten von Uhren: Master und Slaves ➔ Boundary Clock Switches arbeiten gegenüber der Master Clock als Slave ➔ versorgen als Master die weiteren angeschlossenen Slaves
• Hochgenau: PTP benötigt neben dem „normalen“ Ethernetchip noch **extra Hardware** für jeden Teilnehmer ➔ PTP gleicht damit Laufzeiten innerhalb eines Rechnerbetriebssystem aus.

Uhrensynchronisation in verteilten Systemen (4)



Synchronisationsalgorithmus (Quelle Hirschmann)

Uhrensynchronisation in verteilten Systemen (5)



TSU - TimeStamp Unit

- **TSU: Zeitstempeleinheit zum Generieren der Zeitstempel**
  ➔Sync und DelayRequest Nachrichten
- **HW: lokale Uhr**
- **Master/Slave/Switch**

Lokale Rechneruhr (1)

- Anforderung an eine lokale Rechneruhr: **Uhrendisziplin**
  - Zwei aufeinander folgende **Uhrleseoperationen (UOP)** sollen unterscheidbare Zeitwerte liefern
    - ➢ **Systemuhrauflösung sollte signifikant kleiner sein als zwei aufeinander folgende UOPs dauern**

      z. B. GHz Prozessoren mit UOPs im µs-Bereich aber externer Zeitabgleich auf Sekundenbasis.
  - **Eine Uhr soll nur steigende Zeitwerte liefern; d. h. es gibt einen Zeitwert nie zweimal**
    - ➢ Automatisches sprunghaftes **Rückstellen** der Uhr bei Uhrenabgleich ist **nicht zulässig!**

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Lokale Rechneruhr (2)

- **Heutige Realisierungen:** Rechnerinterne Softwareuhr (RSU) wird von einem interrupterzeugenden Hardware-Taktgeber abgeleitet und zählt ein Zählerregister hoch.

- **Beispiel:**
  - Hardware-Taktgeber erzeugt alle **10 ms** einen Interrupt (➔ ***tick*** *-Rate*)
  - OS RSU wird als Zähler in **µs-Auflösung** geführt
    ➔ alle 10 ms wird der ***tick***-Wert ➔ 10 000 zum letzten RSU-Stand aufaddiert.
  - **es sei:** UOP >> 1 µs ➔ Uhrendisziplin 1.: je UOP wird RSU um eins erhöht
  - Zu Systemstartzeit wird die RSU mit der „aktuellen“ Zeit sekundengenau vorgeladen (z.B. CMOS-Uhr)

- **Üblich:**
  Uhrzeit und Datum wird mittels Softwarealgorithmus aus RSU berechnet
  ➔ UNIX-RSU-Basis: Anzahl der Sekunden seit 1.1.1970
  ➔ NTP-Zeitbasis: Anzahl der Sekunden seit 1.1.1900

- **Die lokalen RSUs driften ca. 1 Sekunde in 10 Tagen, d.h. Uhrendriftrate von ca. 10^{-6}.**

Kompensation der Zeitdrift (Uhrenabgleich) (1)

- **Aufgabe**:
  Anpassen der eigenen rechnerinternen Softwareuhr (RSU) an eine erhaltene **Referenzzeit (RZ)** (z.B. UTC via ntp/DCF77-Uhr) unter Wahrung der Uhrendisziplin.

- **Zeitdifferenz (DT), Referenzzeit (RZ) und RSU erfordern Fallunterscheidung**:

  - **Fall 1:**
    Eigene RSU geht nach ➔ „einfaches Vorstellen“ der RSU auf die RZ wäre möglich

  - **Fall 2:**
    Eigene RSU geht vor ➔ Zurückstellen der RSU verboten ! Was tun?

    Idee:
    „Zurückstellen“ der eigenen RSU durch **„Verlangsamen“** bis RZ und RSU synchron sind.
    Die Dauer dieser Anpassung heißt **Anpassungszeit (AT)**.

Kompensation der Zeitdrift (Uhrenabgleich) (2)

Beispiel:

- Gegeben RSU in µs-Auflösung, Timerinterrupt (TI) alle 10 ms, ***tick-***Wert = 10000
- Referenzzeit (RZ) zum Zeitpunkt t_0 empfangen
- ➔ RSU geht um DT = 0,8 Sekunden vor.

➢ **Lösung**
„Verlangsamen" der RSU
Je Timerinterrupt (TI) wird nicht *tick* dazu gezählt, sondern ***tick – tickadj***
wobei ***tickadj*** in der Praxis 1 $^0/_{00}$ – 1 % des *tick*-Wertes ist.

Lösungsbeispiel:
tickadj *= 10* ➔ *verlangsamter tick-Wert während AT :=* ***tick – tickadj = 9990***

- **Zeitdauer der Anpassung**: **AT = DT * *tick/tickadj***

*im Beispiel: AT = 0,8 [s] * tick/tickadj = 1000 * 0,8 [s] =* ***800*** *[s]*

- *statt "einfaches" Vorstellen der RSU ➔ Anpassung durch „beschleunigen" := tick + tickadj*
- *In UNIX existiert der Systemaufruf adjtime() um die RSU auf diese Art und Weise anzupassen*
  ➔ wird von der ntp-Clientsynchronisations-Software benutzt.

- *In UNIX kann zur Systemstartzeit oder mittels manueller Interaktion die RSU auch „hart" gesetzt werden, d. h. ggf.* ***Verletzung der Uhrendisziplin mittels settimeofday()***

Logische Zeit und logische Uhren

- Was ist zu tun wenn es keinen Zugang zu UTC und/oder Zeitserver gibt oder der Aufwand zu hoch ist?

- Beobachtung:
  Für die Feststellung der zeitlichen Reihenfolge von Ereignissen ist oft die Ordnung (Nummerierung) der Ereignisse entlang ihres Entstehens ausreichend.
  ➔ **physikalische Zeit nicht nötig!**

- „Nummerierung“ der Ereignisse führt zum Konzept einer logischen Uhr, bzw. einer logischen Zeit auf Basis einer „globalen“ Zählvariable:

  - geringerer Realisierungsaufwand als exakte physikalische Zeit
  - an die aktuellen Anforderungen angepasste „Auflösung“, pro Ereignis ein Zahlenwert
  - Nachteil: **keinen Zeitbezug zur physikalischen Zeit**

- **Aber:** Wie funktioniert die Idee der logischen Zeit – eineindeutige Nummerierung der Ereignisse - im verteilten System??

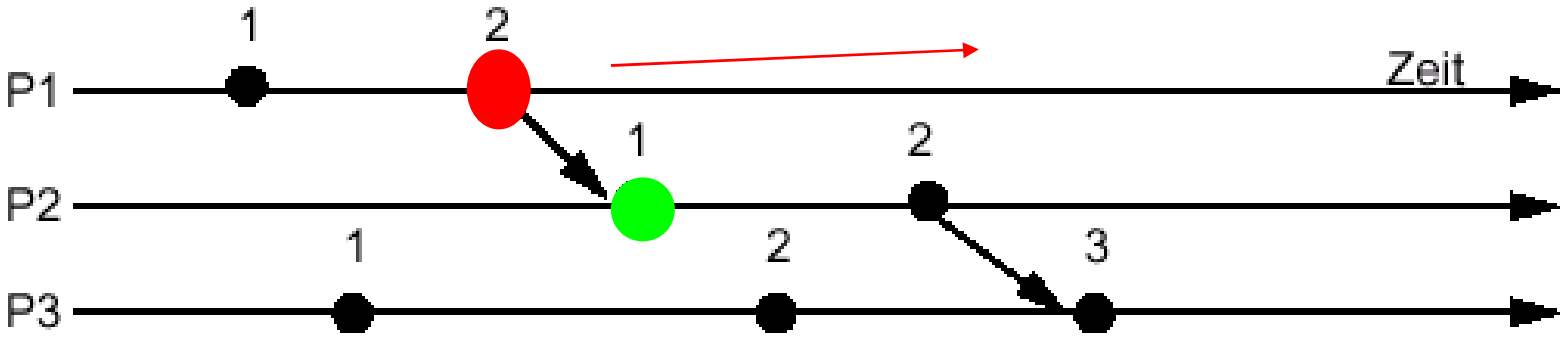
- **Problem**: Nebenläufigkeit der Prozesse laufend auf verschiedenen Rechnern **und** keine „globale“ Zählvariable verfügbar.

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Lamport-Zeit (1)

- Happened-before-Relation als Bildungsgesetz der logischen Zeit
  - Definition: **Happened-before**
    - *Sind a und b Ereignisse im gleichen Prozess und a ereignet sich vor b, dann gilt: a ➔ b (a happened before b).*
    - *Ist a ein Sendeereignis in einem Prozess und b das zugehörige Empfangsereignis in einem anderen Prozess, dann gilt: a ➔ b*
      (Endliche Zeit zur Nachrichtenübermittlung nötig!)
  - **Happened-before Relation entspricht der kausalen Ordnung von Ereignissen.**
    - Transitivität, d.h. wenn *a* ➔ *b* und *b* ➔ *c* gilt, dann gilt auch *a* ➔ *c*.
    - unabhängige Ereignisse (es gilt weder *a* ➔ *b* noch *b* ➔ *a*) heißen nebenläufig (engl.: concurrent) und werden als *a||b* geschrieben
  - Happened-before definiert nur eine **partielle** (=**teilweise**) Ordnung auf Ereignisse!

Lamport-Zeit (2)

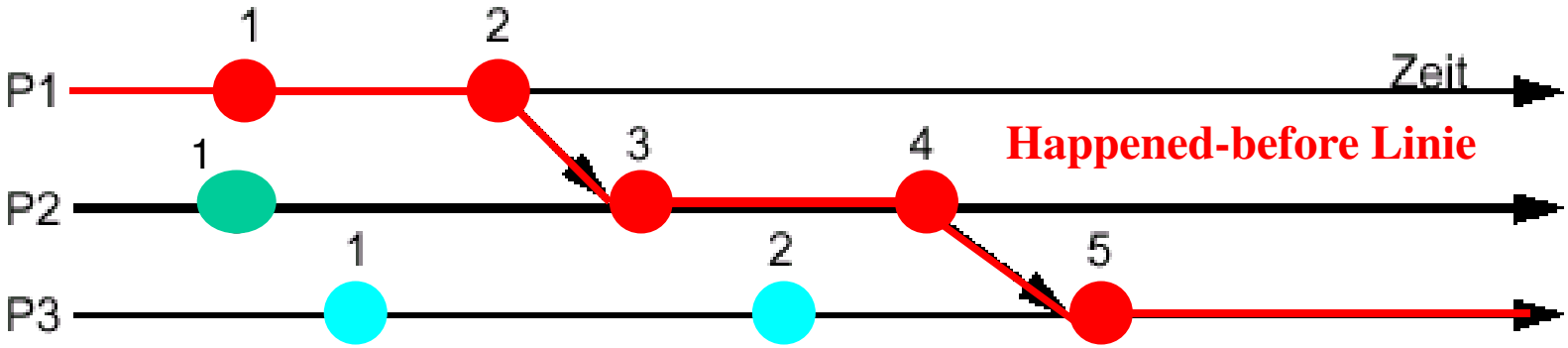


• ohne Lamport Synchronisation

- **Logische Uhr nach Lamport** (1978)
  - jeder Prozess führt eine eigene sog. logische Uhr C_P als eine Variable vom Typ natürlicher Zahlen
  - Bei jedem Ereignis a wird die Uhr C_P verändert, und das Ereignis a mit dem neuen Zeitstempel C_P versehen ➔ $C_P(a)$ bezeichnet den Zeitstempel des Ereignisses a.
  - Die logische Uhr soll der **Happened-before-Relation** genügen.

Lamport-Zeit (3)

- **Lamport-Algorithmus:**
  - *Initialisierung = erstes lokales Ereignis im Prozess P* ➔ $C_P := 1$;
  - *Sendeereignis S im Prozess P* ➔
    $C_P := C_P + 1$;
    send (message, C_P); ➔ $C_p(S)$
  - *Empfangsereignis E im Prozess Q* ➔
    (mit Sendeprozess P)
    receive (message, C_P);
    $C_Q := \mathbf{max(C_Q, C_P) + 1}$;
    ➔ $C_Q(E)$
  - *lokales Ereignis a im Prozess P* ➔ $C_P := C_P + 1$; ➔ $C_P(a)$



• mit Lamport Synchronisation

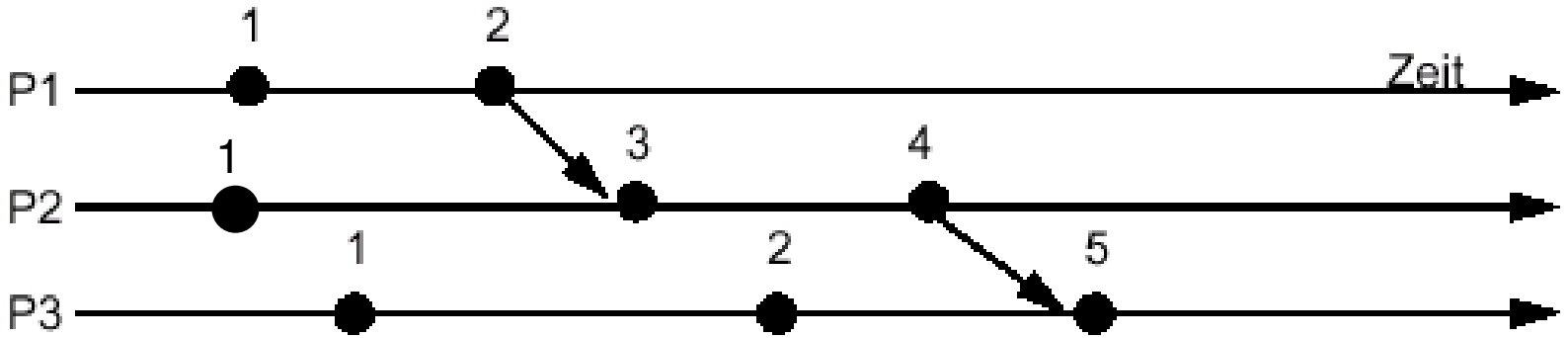
Lamport-Zeit (4)

- **Erweiterung der Lamport-Zeitdefinition zu einer willkürlichen globalen totalen Ordnung:**
  Für alle Ereignisse a,b gilt nun global (über alle Prozesse) **per Definition:**
  - wenn C(a) < C(b) ➔ $a \rightarrow b$ ('<' = 'numerisch kleiner')
    - ggf. Aufhebung der tatsächlichen Ordnung:
      Im Bild ➔ P2 Ereignis 3 ist **nun** nach P3 Ereignis 2, eine tatsächlich aufgetretene Nebenläufigkeit wird willkürlich sortiert.

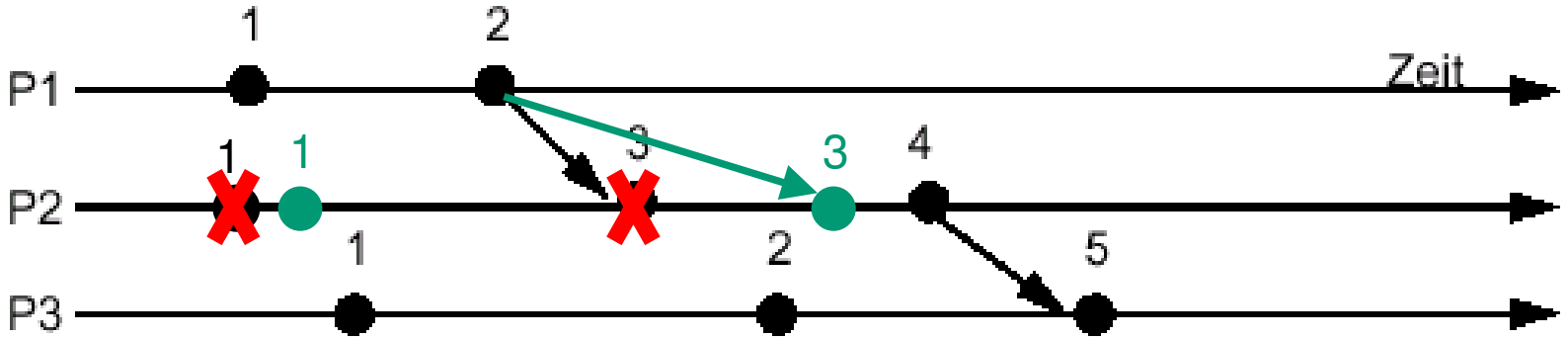
  - wenn C(a) = C(b) dann wird künstlich C(a) < C(b) herbeigeführt
    - durch Berücksichtigung eines weiteren **globalen allen gleich bekannten numerischen** Unterscheidungsmerkmal z.B. C().<Rechnernr>.<ProzessID>
    - bei Prozessen des gleichen Rechners reicht z. B. C_p.<Prozess-ID> aus.
      z. B.
      Prozess 1 Zeitstempel bei Ereignis a := $C_{P1}(a).1$
      Prozess 2 Zeitstempel bei Ereignis b := $C_{P2}(b).2$
    - Beispiel im Bild wenn $C_{P1}(a) == C_{P2}(b) = 1$
      ➔ **dann** ist trotzdem $C_{P1}.1 < C_{P2}.2$ und damit gilt global **per Definition** willkürlich:
      Ereignis a in P1 vor (!!) b in P2

Lamport-Zeit (4)

Darstellung der tatsächlichen Ordnung:

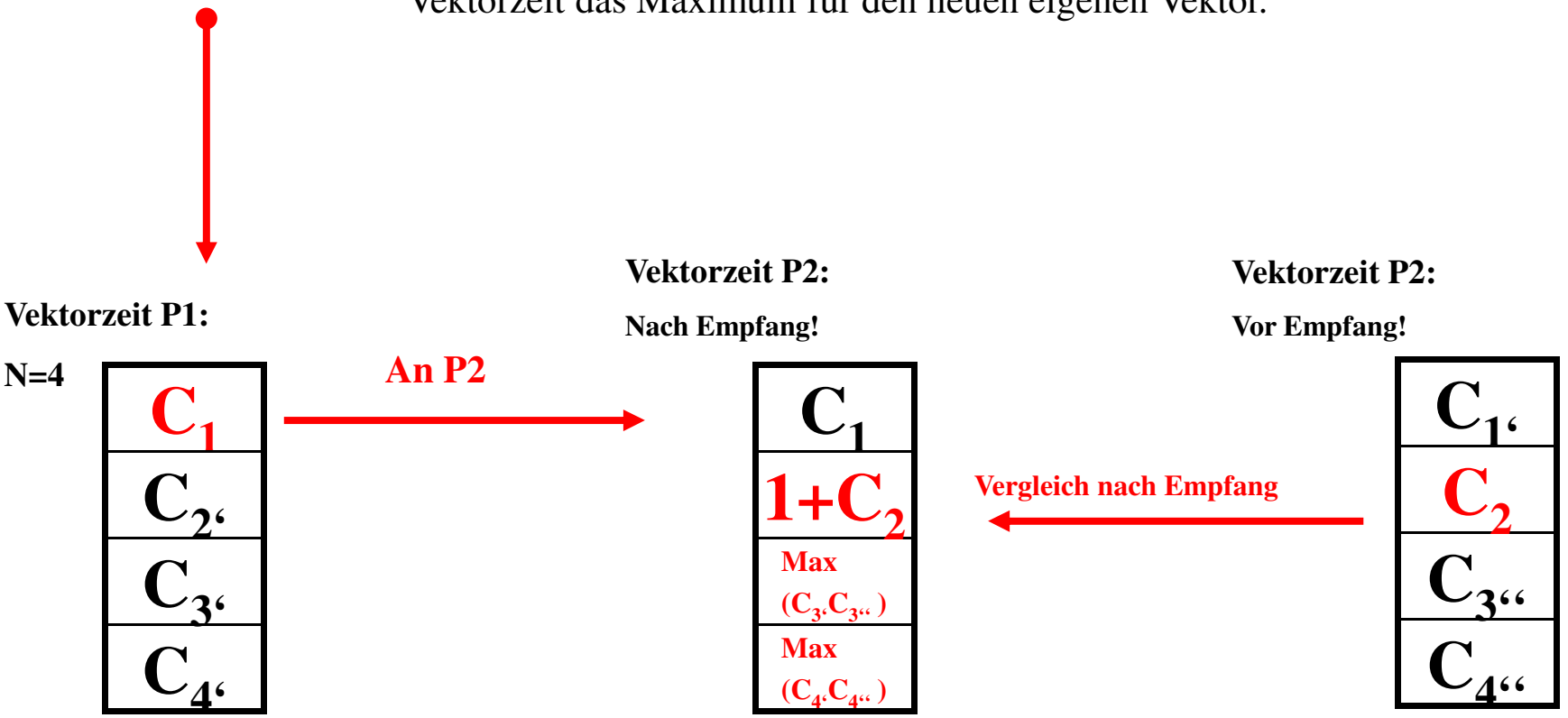


Nach Anwendung der willkürlichen totalen Ordnung:



Vektorzeit (1)

- Sendeereignis in Pi
  ➔ Verschicken der eigenen Vektorzeit an Empfänger Pj
- Empfänger bildet komponentenweise aus eigener und empfangener Vektorzeit das Maximum für den neuen eigenen Vektor.



Vektorzeit (2)

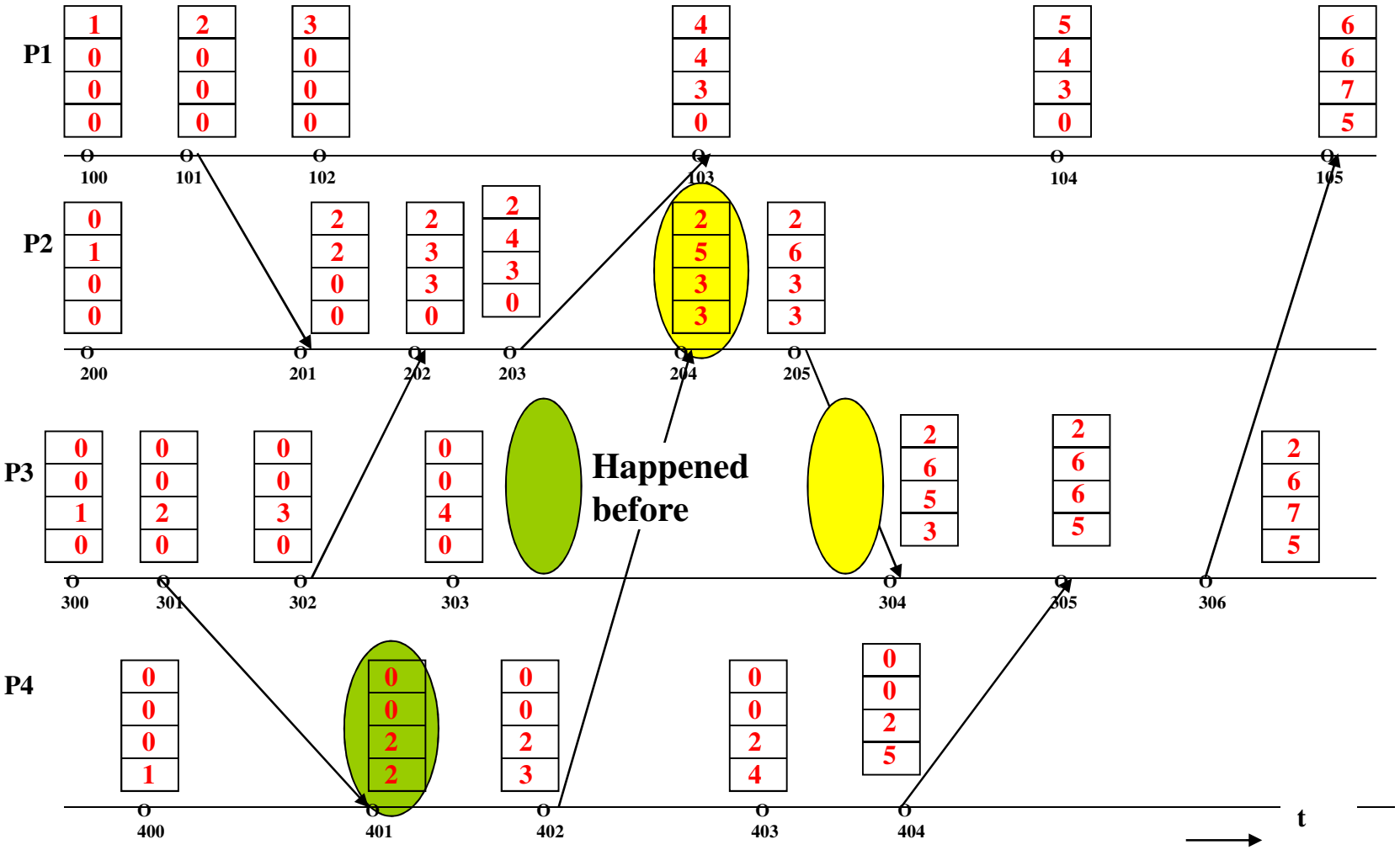


Bild S.3 Vektorzeitbeispiel

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Vektorzeit (3)

- **Eigenschaften:**
  - Ein externer Beobachter hat die Möglichkeit fremde Ereignisse zu ordnen, denn anhand der Zeitstempel zweier Ereignisse lässt sich ihre Kausalitätsbeziehung erkennen.
  - Vektorzeit ist kausalitätserhaltend ➔ d. h. aus a **happened before** b folgt VC(a)<VC(b).
  - **Die Umkehrung der Uhrenbedingung gilt ebenfalls**
    ➔ d. h. aus VC(a)<VC(b) folgt a happened before b.
- **Anwendungen :**
  - Kausal geordnete Gruppen- und Einzelkommunikation
    - Schiper-Eggli-Sandoz (SES) Allg.: kausale Ordnung von Einzelmessages
    - Birman-Schiper-Stephenson (BSS) Allg.: kausale Ordnung von Broadcasts
  - Verteilter gemeinsam genutzter Speicher (shared memory).
  - Implementierung der Konsistenz von „Checkpoints“ in „optimistic recovery“