Einführung Echtzeitsysteme

Echtzeitsysteme im Alltag

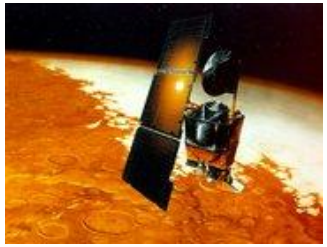
Echtzeitsysteme sind allgegenwärtig!







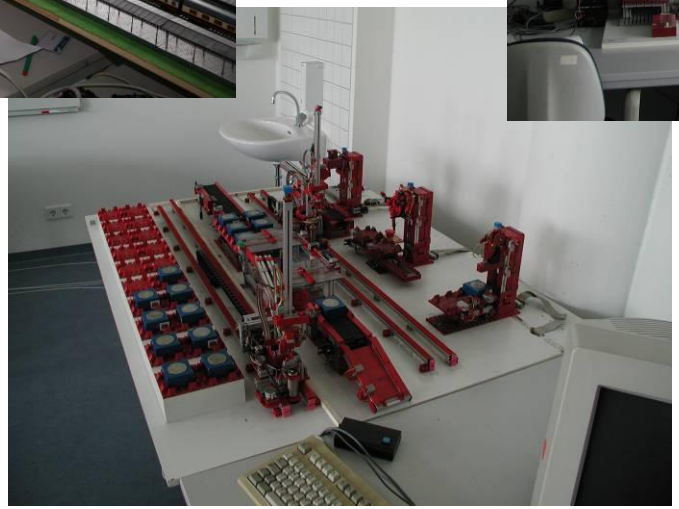




Einleitung Echtzeitsysteme

Anwendungen am Lehrstuhl / fortiss

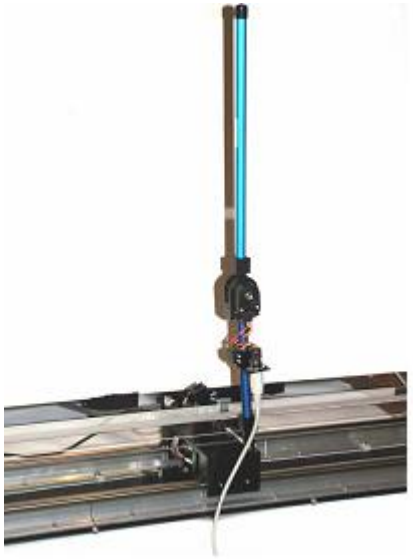
Steuerungsaufgaben (Praktika+Studienarbeiten)



Regelungsaufgaben (Praktika+Studienarbeiten)

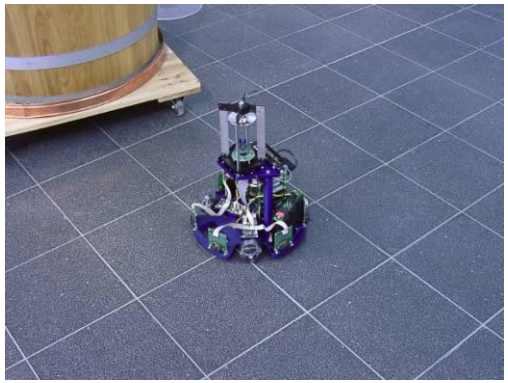
Schwebender Stab

Produktionstechnik

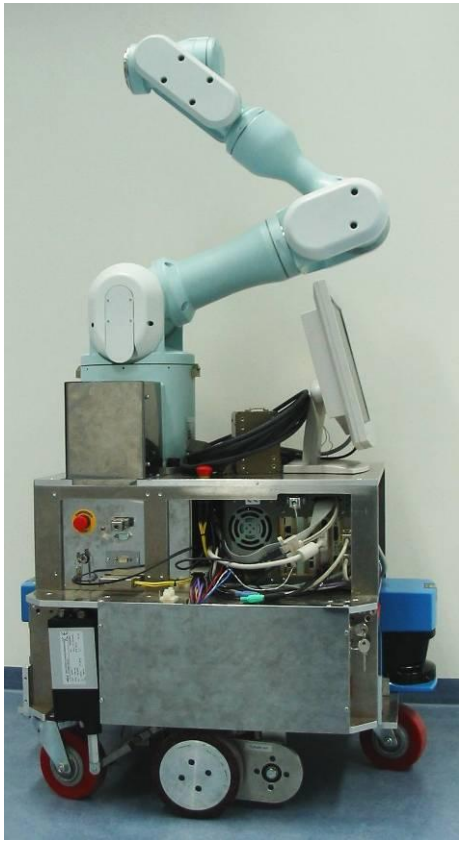


Invertiertes Pendel

Robotersteuerung

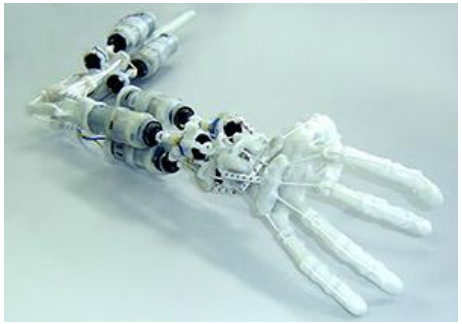


Robotino



Leonardo

Stäubli



Tumanoid

Aktuelle Entwicklung: Cyber-Physical Systems

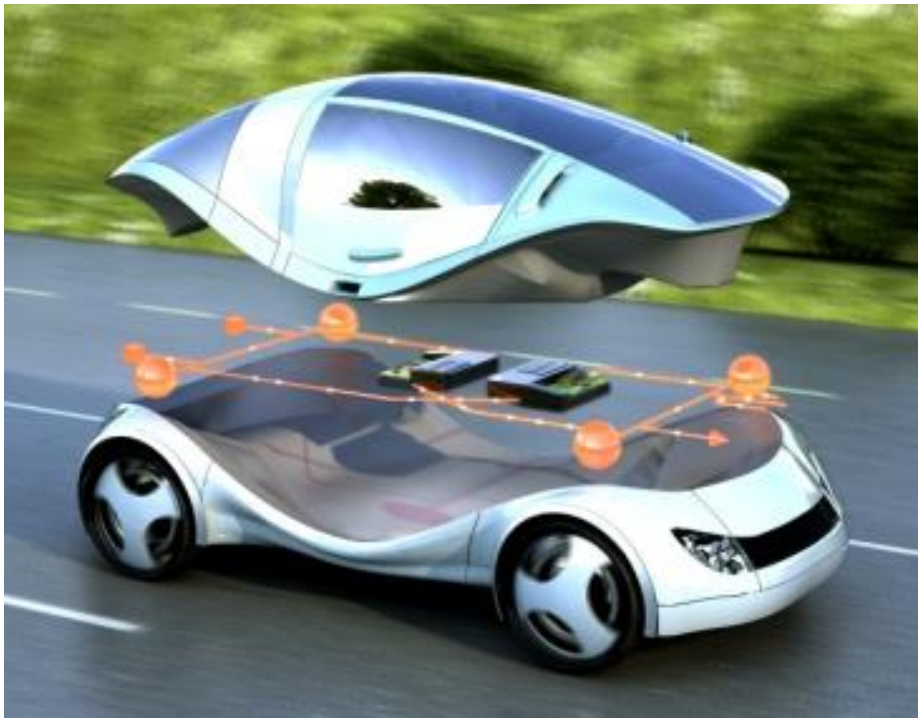
- Cyber-Physical System (CPS) bestehen aus mehreren Subsystemen von denen mindestens eines dieser Subsysteme mit der Umgebung interagiert (→ eingebettetes System mit Echtzeitanforderungen). Die Subsysteme sind typischerweise unabhängig voneinander entwickelt worden, sollen aber doch die Integration miteinander eine bessere Funktionalität liefern. Wandel ist ein ständiger Begleiter.
- Abgrenzung zu anderen Forschungsthemen:
  - Aktive Interaktion mit der Umgebung: Sensornetzwerke sind zunächst keine CPS
  - Unabhängigkeit der Subsysteme: Unterscheidung zu "klassischen" eingebetteten, vernetzten Systemen
  - Generell: die Integration muss sehr häufig autonom erfolgen, es gibt keine Firma, die die Integratorenrolle übernimmt → Integration über Standards anstelle durch Personen
- Mehr zu dem Thema unter: http://www.acatech.de/?id=1405 (Agenda CPS)

Leitprojekte von fortiss im Bereich Echtzeitsysteme / Cyber-Physical Systems

- Am fortiss wird eine große Zahl von Projekten im Bereich Echtzeitsysteme bzw. Cyber-Physical Systems für die Anwendungsgebiete Automotive, Industrieautomatisierung und Robotik durchgeführt
- Nahezu alle Projekte werden zusammen mit attraktiven Industriepartnern durchgeführt (u.a. Audi, BMW, EADS, Festo,Siemens)
- Für die Hörer dieser Vorlesung bieten sich zum Einstieg drei Projekte besonders an: die Leitprojekte "**RACE**", "**AutoPnP**" und "**SME Robotics**"
- Wir freuen uns über interessierte Studenten und bieten:
  - Spannende Studienarbeiten (Bachelor-, Master-, Guided Research)
  - Interessante Werkstudententätigkeiten
  - Möglichkeiten zur Promotion als wissenschaftlicher Mitarbeiter
- Mehr unter: http://www.fortiss.org/karriere/ueberblick/

Projektidee RACE

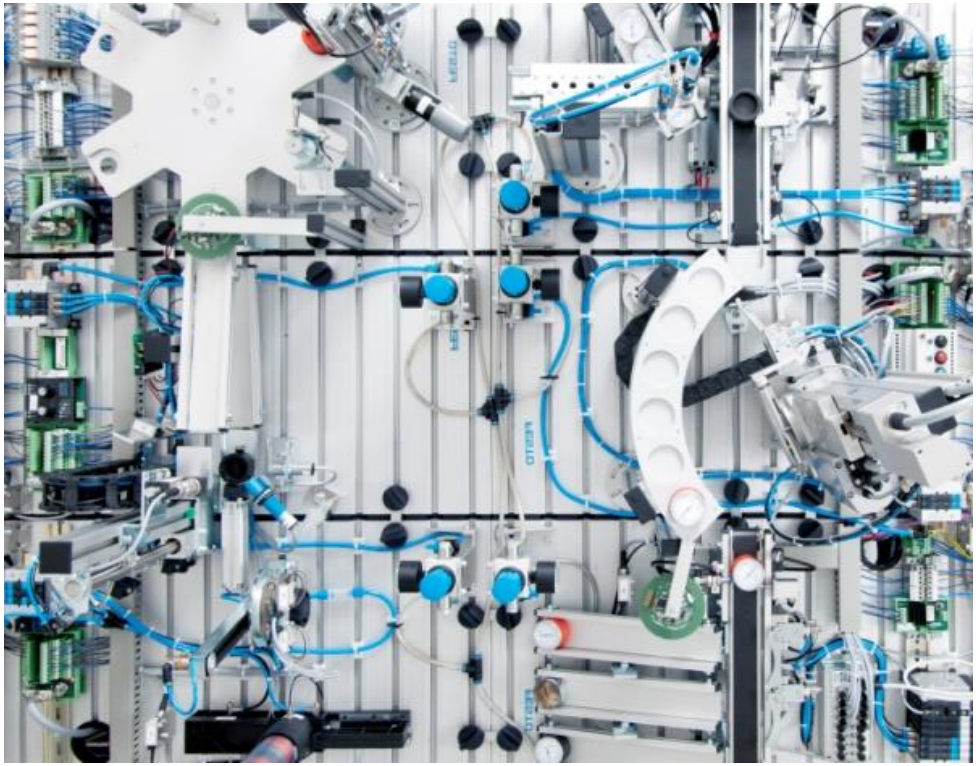
- Entwicklung eines Versuchsträgers für Plug&Play-fähige Fahrzeuge auf Basis eines Zentralrechnerkonzeptes
- Deutliche Vereinfachung der Entwicklung auf von komplexen Funktionen, wie dem autonomen Fahren
- Im Rahmen eines Studentenwettbewerbes können Studenten eigene Funktionen für das neue Elektrofahrzeug entwickeln
- Mehr Infos unter
  www.fortiss.org/ikt2030
  www.race-projekt.de



RACE Demonstrator – Autonomes Fahren

Projekt AutoPnP – Industrie 4.0

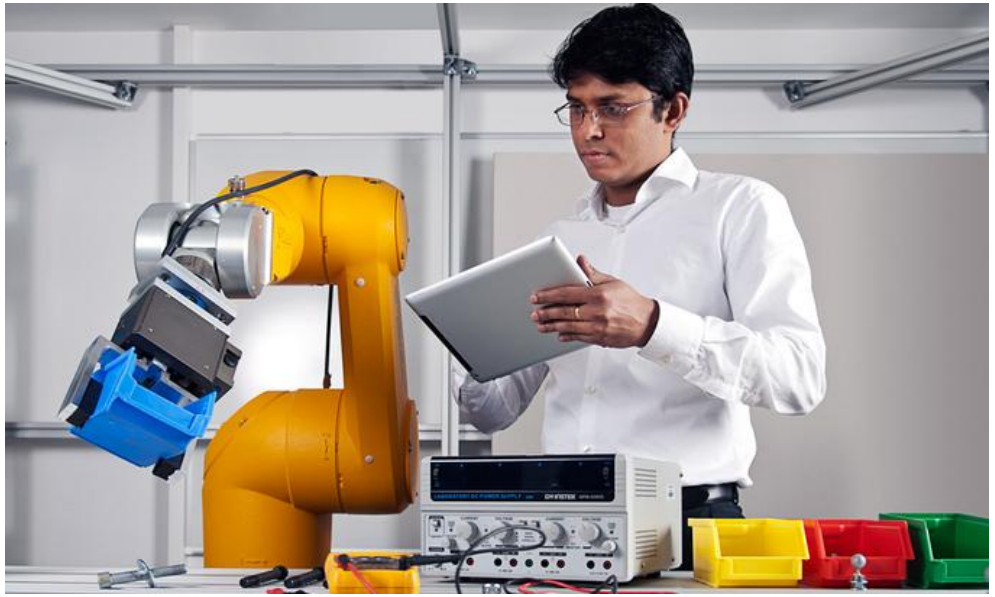
- Entwicklung einer wandelbaren Fabrik, die vollautomatisch ihre Topologie erkennt und das Produktionsprogramm synthetisiert und so sehr flexibel an neue Produkte angepasst werden kann.



Mehr Infos unter
http://www.fortiss.org/forschung/projekte/autopnp/

Projekt SME Robotics

- Entwicklung von intuitiv programmierbaren Robotern für den produzierenden Mittelstand



- Mehr Infos unter http://www.fortiss.org/forschung/projekte/smerobotics-1/

Erfolgskontrolle: Was sollten Sie aus dem Kapitel mitgenommen haben?

- Definition und Unterscheidung der Begriffe:
  - Echtzeitsystem und eingebettete Systeme
  - Harte / weiche Echtzeitsysteme
- Kenntnis von Beispielen für Echtzeitsysteme
- Wichtigste Aussage: Echtzeitsysteme heißt nicht schnell, sondern die Garantierung von maximalen Antwort- / Berechnungszeiten

  → Grundlage zur Beurteilung einer Technologie, ob diese echtzeitfähig ist

Klausurfragen

- Klausur WS 06/07
  - Was ist der Unterschied zwischen harten und weichen Echtzeitsystemen? (3 Punkte = 3 min)
    - **Antwort**: siehe Folien
  - Wieso sollte Virtual Memory nicht in Echtzeitsystemen verwendet werden? (3 Punkte = 3 min)
    - **Antwort**: Bei Verwendung von Virtual Memory hängt die Ausführungszeit davon ab, ob bei einem Speicherzugriff der Speicherplatz schon im Virtual Memory ist oder erst aus dem Hintergrundspeicher nachgeladen werden muss → die Zugriffszeiten unterscheiden sich für beide Situationen deutlich und erschweren die Zeitanalyse.
- Wiederholungsklausur WS 06/07 (5 Punkte = 5 min)
  - Ordnen Sie folgende Anwendungen in die Kategorien harte bzw. weiche Echtzeitsysteme ein und begründen Sie Ihre Antwort:
    - Ampelsteuerung
    - Flugzeugregelung
    - Internettelefonie
  - **Antwort**: Harte Echtzeitsysteme: Flugzeugregelung (Verzögerung kann zu Absturz führen), Ampelsteuerung (durch verzögerte Nachrichtenübermittlung können zwei Ampeln gleichzeitig auf grün schalten → Unfallgefahr), weiche Echtzeitsysteme: Internettelefonie (Verzögerungen führen zu Aussetzern im Gespräch → verminderte Qualität, aber keine großen Schäden); Zeitablauf einer Ampel: wenn die Ampel länger rot bleibt (aber das Gesamtsystem konsistent ist) führt die zu einer verminderten Qualität

Klausurfragen

- Klausur WS 10/11
- Gegeben sind folgende Aussagen über Echtzeitsysteme:
  - (i) Harte Echtzeitsysteme sind Systeme, die besonders schnell sind, also besonders kurze Berechnungszeiten haben.
  - (ii) Harte Echtzeitsysteme sind Systeme, bei denen ein bestimmtes Zeitverhalten garantiert werden kann, wobei das Zeitverhalten aber langsam sein kann.

- Welche dieser Aussagen ist richtig? Geben Sie für die richtige Aussage ein Beispiel an, das ein Echtzeitsystem ist. Geben Sie für die falsche Aussage ein Beispiel an, das zwar die Aussage erfüllt, aber kein Echtzeitsystem ist.
- **Antwort**:
  - Aussage (i): Falsch. Beispiel: eine Anfrage in einer Suchmaschine wird zwar sehr schnell beantwortet, allerdings ist es kein hartes Echtzeitsystem.
  - Aussage (ii): Richtig. Beispiel: Ampelsteuerung. Das Zeitverhalten ist im Sekunden- bzw. Minutenbereich, allerdings muss sichergestellt sein, dass alle Ampeln gleichzeitig umschalten.

Kapitel 2

Uhren & Synchronisation

Inhalt

- Motivation
  - Definition Zeit
- Uhren
- Synchronisation
  - Algorithmus von Cristian
  - Algorithmus aus Berkeley
  - NTP-Protokoll
  - Synchronisation bei fehlerbehafteten Uhren

Literatur

- Links zum Thema Zeit:
  - http://www.ptb.de/de/zeit/uhrzeit.html
  - http://www.maa.mhn.de/Scholar/dt_times.html
- Uhrensynchronisation:
  - Leslie Lamport: Synchronizing clocks in the presence of faults, 1985
  - http://www.ntp.org/

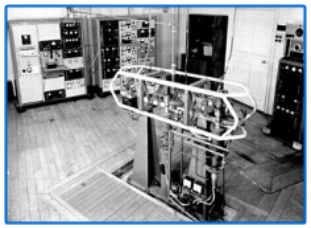
Relevanz für Echtzeitsysteme: Zeit und Ordnung

- Da viele Vorgänge in der Umwelt und Physik auf Zeit basieren, müssen Echtzeitsysteme häufig die Zeit wahrnehmen und darauf basierend interagieren können.
- Weitere Anforderung: Einordnen von Ereignissen im verteilten System (bevorzugt im Sinne einer zeitlichen Abfolge) ➔ globale Zeitbasis hilft bei der Ableitung einer zeitlichen Ordnung ➔ Zeitsynchronisation wird benötigt

Definition Zeit

- Historisch:
  - Jeden Tag gegen Mittag erreicht die Sonne ihren höchsten Punkt am Himmel.
  - Die Zeitspanne zwischen zwei aufeinander folgenden Ereignissen dieses Typs heißt Tag (genauer gesagt: ein Sonnentag).
  - Eine Sonnensekunde ist 1/86400 dieser Spanne.
- Zeitmessung heute:
  - Verwendung von Atomuhren: eine Sekunde ist die 9.192.631.770-fache Periodendauer, der dem Übergang zwischen den beiden Hyperfeinstrukturniveaus des Grundzustands von ^{133}Cäsium-Atomen entsprechenden Strahlung.
  - Am 01.01.1958 entsprach die Atomsekunde genau einer Sonnensekunde.
  - Aufgrund von unregelmäßigen Schwankungen, sowie einer langfristigen Verlangsamung der Erdrotation unterscheiden sich die Atomsekunde und die Sonnensekunde.

Sonnenuhr
Deutsches Museum



Erste Cäsiumatomuhr

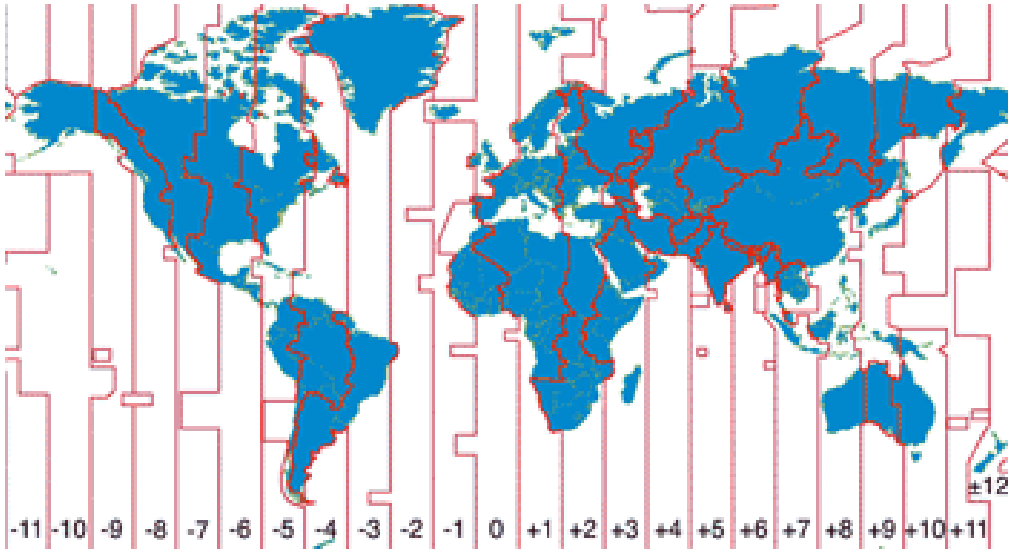
TAI (Temps Atomique International)

- TAI: Atomzeitskala, die zur Koordination nationaler Atomzeiten ermittelt wird:
  - Beteiligung von 50 verschiedene Zeitinstitute mit ca. 250 Atomuhren
  - Zeit basiert auf der Atomsekunde
  - Referenzzeitpunkt ist der 1.Januar 1970
  - relative Genauigkeit von +/ - 10^{-15}, aber keine exakte Übereinstimmung mit der Sonnenzeit

Atomuhr der Physikalisch-Technischen Bundesanstalt in Braunschweig

UTC (Coordinated Universal Time)

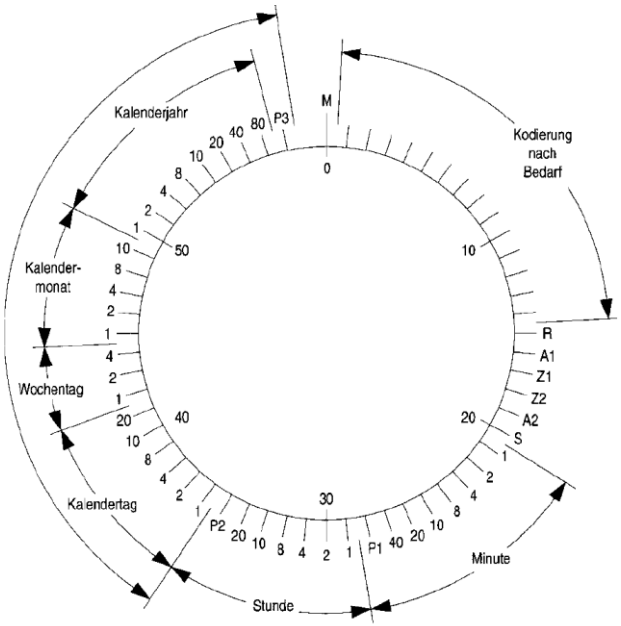
- eigentlicher Nachfolger der Greenwichzeit
- realisiert durch Atomuhren, die Zeiteinheit ist die SI-Sekunde
  → hochkonstante Zeiteinheit
- zusätzlich Übereinstimmung mit dem Sonnenlauf
  → einheitliche Grundlage zur Zeitbestimmung im täglichen Leben
- Durch Einfügen von Schaltsekunden wird die UTC mit der universellen Sonnenzeit (UT1) synchronisiert
- Anpassung erfolgt zumeist zu Ende oder Mitte des Jahres (typischer Abstand: alle 18 Monate)



Zeitzonen

DCF77

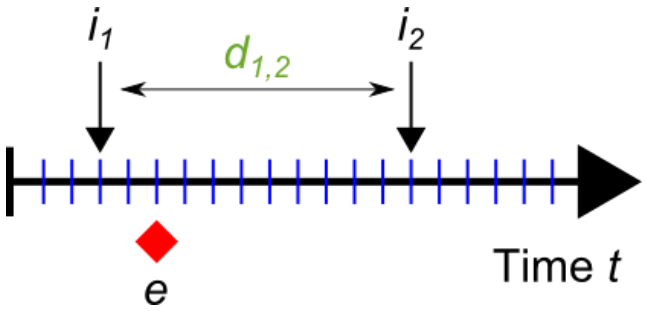
- Das PTB überträgt die aktuelle Uhrzeit über den Langwellen-sender DCF77
- Die Zeitinformationen werden als digitales Signal (negative Modulation → Absenkung der Trägeramplitude) im Sekunden-takt übertragen.



- '0' und '1' werden durch eine Absenkung um 100ms bzw. 200 ms codiert. In der Sekunde 59 erfolgt keine Absenkung → Markierung der Beginn einer neuen Minute bei nächster Amplitudenabsenkung.
- Pro Minute stehen somit 59 Bit zur Verfügung (wobei Bit 0-14 für Betriebsinformationen verwendet werden)

Zeit als Modell

- Zeit kann durch unendliche Menge T von Zeitpunkten („Instants“) modelliert werden
  - T ist dann eine geordnete Menge, d.h. falls p und q zwei Zeitpunkte sind so treten sie entweder:
    - Gleichzeitig oder
    - Eines nach dem anderen ein → p vor q oder q vor p (mutually exclusive)
  - Ordnung zwischen den Zeitpunkten heißt temporale Ordnung
  - T ist dicht
    - zwischen zwei Zeitpunkten p und r liegt zumindest ein weiterer Zeitpunkt q, dann und nur dann wenn p und r nicht gleichzeitig sind.
  - Intervall aus T heißt Zeitraum oder Dauer



Uhren und Synchronisation

Uhren

Aufgaben

- Absolutzeitgeber
  - Datum, Uhrzeit
  - zeitabhängige Aufträge
  - Zeitstempel, Logbuch
  - Ursache-Wirkung-Feststellung
- Relativzeitgeber
  - Verzögerungen
  - Messen von Zeitabständen
  - Zyklische Ausführung, Messungen
  - Zeitüberwachung von Wartezuständen





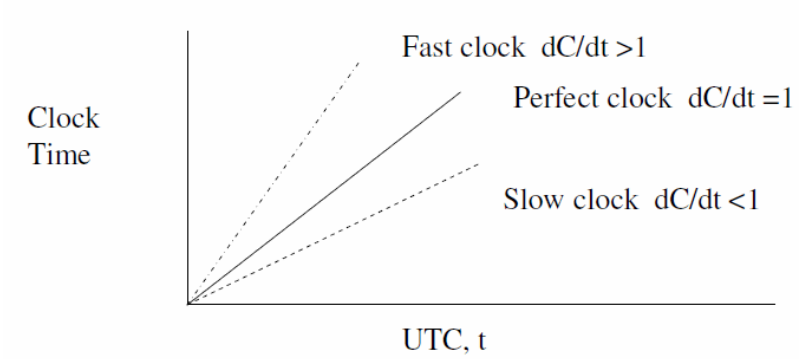
Genauigkeit von Uhren

- Eine Uhr arbeitet korrekt, wenn sie die vom Hersteller angegebene maximale Driftrate τ einhält, auch wenn sie dann etwas zu schnell oder zu langsam ist.
- Typische Driftraten:

Uhrentyp	**Driftrate τ**	**Abweichung pro Jahr**
Quarzuhr	10^{-5}	~ 300 sec
Pendeluhr	10^{-6}	~ 30 sec
Atomuhr	$1{,}5*10^{-14}$	~ 0,5 Mikrosekunden
Atomuhr (lasergekühlte Atome)	10^{-15}	~ 0.03 Mikrosekunden

Uhrenverhalten

- Korrekt:
  1. Absolutwert der Abweichung kleiner der zugesicherten Gangabweichung
- Fehlerbehaftet:
  2. Überschreiten der zugesicherten Gang-abweichung
  3. Zustandsfehler (z.B. Sprung im Zählerwert)
  4. Stehenbleiben der Uhr
- Unmöglich:
  5. Rückwärtslaufende Uhr
  6. Unendlich schnell laufende Uhr



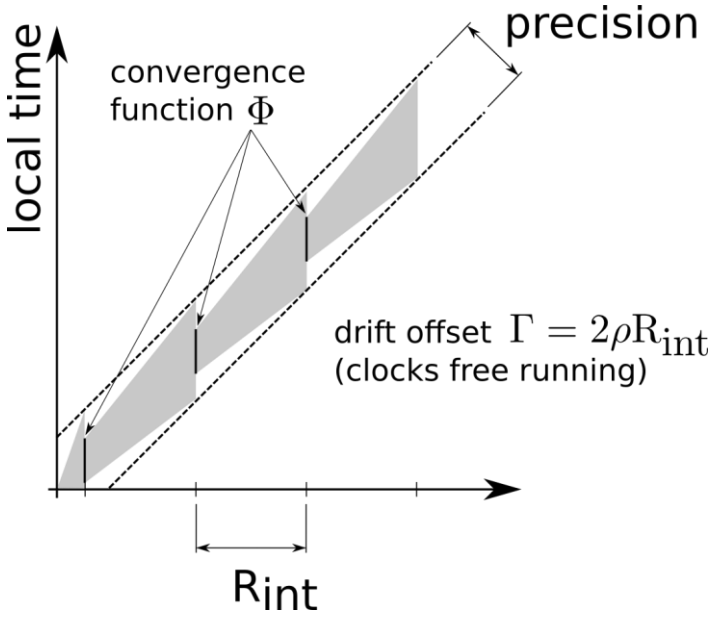
➔ Die Gangabweichung zweier korrekter Uhren kann beliebig groß werden, wenn die Uhren nicht synchronisiert sind.

Uhren und Synchronisation

Synchronisation

Ziele der Uhrensynchronisation

- Ziel ist es verschiedenen, unabhängigen Prozessen, die auf unterschiedlichen Rechnern laufen, zu ermöglichen konsistente Entscheidungen über die Ordnung verschiedener Ereignisse zu ermöglichen
- Unterscheidung zwischen:
  - Logischer Uhr: Zuordnung von Sequenznummern zu Ereignissen
  - Physikalische Uhren: Zuordnung eines Zeitstempels zu Ereignissen
- Im Rahmen dieses Abschnittes wird nur die Synchronisation von physikalischen Uhren betrachtet. Mehr zur Synchronisation von logischen Uhren findet man unter: http://www.cs.rutgers.edu/~pxk/417/notes/content/08-clocks.pdf



Grundlagen

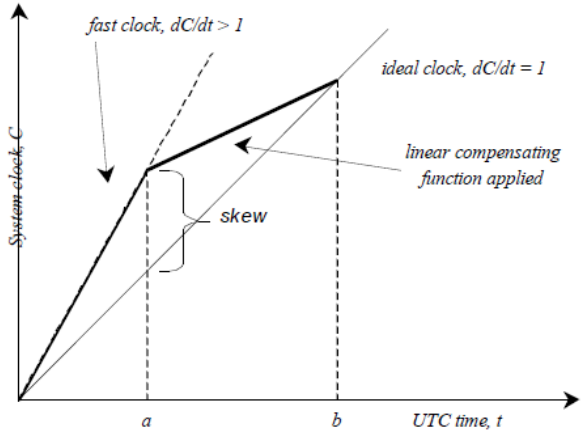
- Folgende Annahmen werden im Zusammenhang mit der Synchronisation gemacht:
  1. Alle Uhren besitzen zu Beginn in etwa die gleiche Zeit (in diesem Abschnitt wird vor allem die laufenden Synchronisation betrachtet).
  2. Die Uhren fehlerfreier Prozesse gehen annähernd richtig, d.h. sie besitzen eine ähnliche Ganggenauigkeit.
  3. Ein fehlerfreier Prozess *p* kann die Differenz seiner Uhr von der Uhr von Prozess *q* mit einer Genauigkeit ε bestimmen.
- Anforderungen an die Synchronisation:
  1. Zu jedem Zeitpunkt zeigen die Uhren zweier fehlerfreier Prozesse ungefähr den gleichen Wert.
  2. Durch die Synchronisation entstehen keine bzw. nur sehr kleine Zeitsprünge
  3. Insbesondere darf die Kausalität nicht verletzt werden (z.B. Zurückstellen der Zeit)

  → Notwendig, da sonst keine konsistente Ausführung (z.B. wegen Anweisungen mit absoluten Zeitangaben) garantiert werden kann.

Arten der Synchronisation

- Zeitpunkt: typischerweise erfolgt die Synchronisation periodisch
- Rollen der Knoten:
  - externe Synchronisation: die Synchronisation erfolgt anhand einer externen, als perfekt angenommenen Uhr
  - interne Synchronisation: die Uhren ermitteln basierend auf den einzelnen Zeitwerten eine korrekte, globale Zeitbasis

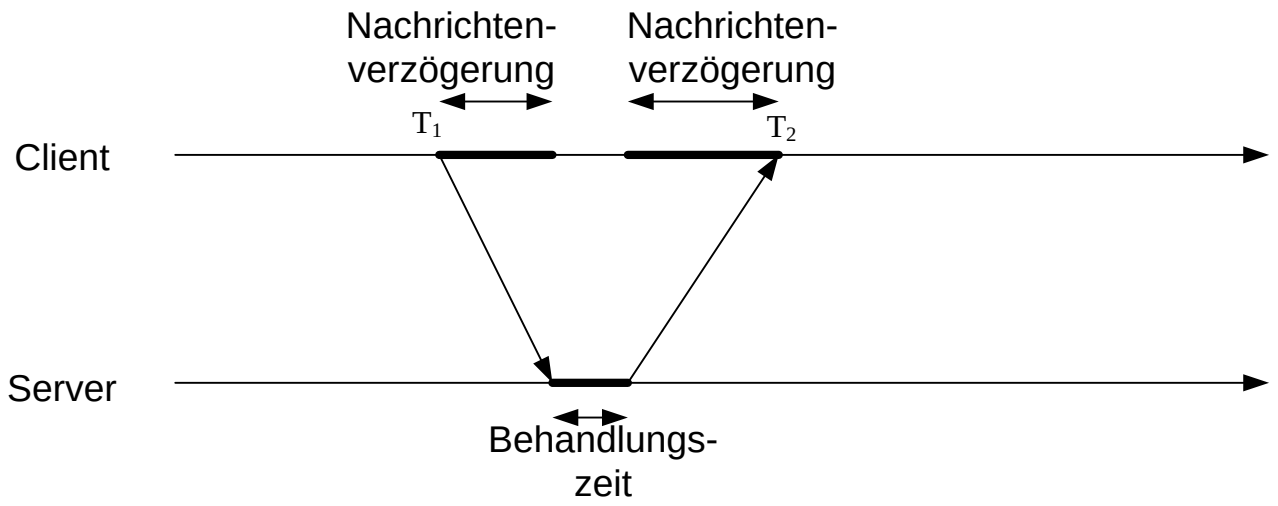
  Vorteil der externen Synchronisation: der maximal tolerierte Fehler kann halb so groß wie bei der internen Synchronisation gewählt werden.
- Ort der Synchronisation:
  - zentrale Synchronisation: Synchronisation wird von einer Einheit koordiniert → fehleranfällig
  - verteilte Synchronisation: alle Einheiten berechnen die globale Zeitbasis → hohes Datenaufkommen
- Umgang mit Synchronisationsunterschieden:
  - Anpassung der Uhrenfrequenz
  - Kleine Zeitsprünge: wichtig ist, dass hier Probleme mit der Programmlogik ausgeschlossen werden



© Krzyzanoski, Rutgers University, CS 417: Distributed Systems

Algorithmus von Cristian (1989)

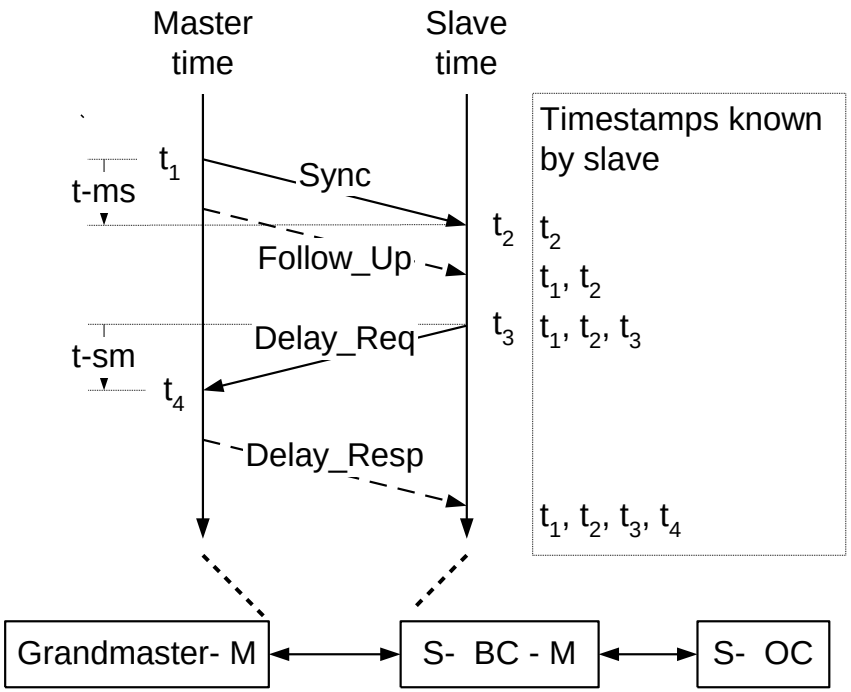
- Das Verfahren basiert auf verteilter, externer Synchronisation.
- Innerhalb des Systems existiert ein Time-Server, zumeist ein UTC-Empfänger.
- In regelmäßigen Abständen senden die anderen Einheiten einen Time-Request, der so schnell wie möglich vom Server beantwortet wird.



Algorithmus von Cristian

- Nach Empfang kann die Uhr auf die empfangene Uhrzeit gesetzt werden.
- **1. Problem**: Zeitsprünge würden entstehen.
- **Lösung**: Die Uhr wird graduell angepasst (Beispiel: Herabsetzung des Intervalls zwischen zwei Uhrenticks von 1ms auf 0.9ms, falls lokale Uhr zu langsam war).
- **2. Problem**: Nachricht ist veraltet, wenn die Nachrichtenverzögerung nicht vernachlässigbar ist.
- **Lösung**: Messung der Nachrichtenverzögerung
  - Abschätzung, falls Informationen fehlen: $(T_1-T_2)/2$
  - Falls die Behandlungszeit bekannt ist, kann das Ergebnis weiter verbessert werden.
  - Zusätzliche Verbesserung: Ermittlung eines Durchschnittswertes, Ausreißer müssen dabei außer acht gelassen werden.
  - Umso genauer die Verzögerung berechnet werden kann, desto besser funktioniert die Uhrensynchronisation

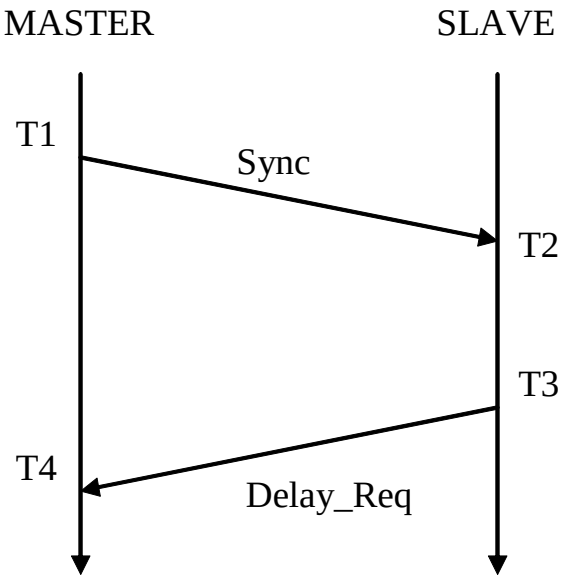
Erhöhung der Genauigkeit: Standard IEEE 1588

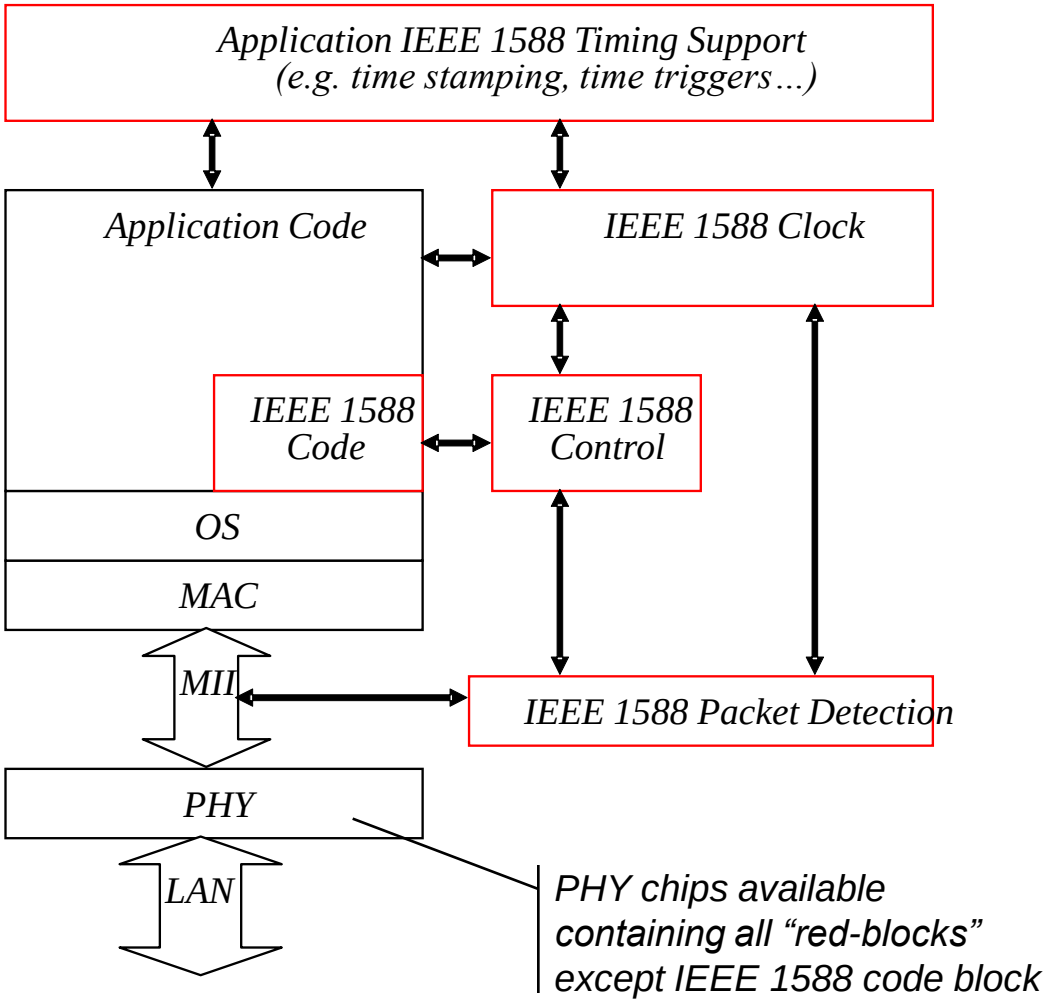


© John Eidson, Vortrag 09.10.2009, München

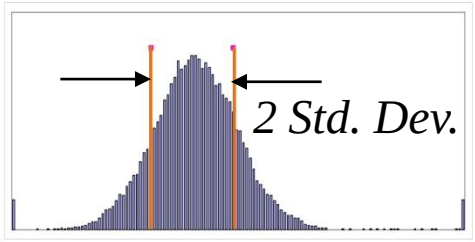
- Under the assumption that the link is symmetric
- Offset = (Slave time) – (Master time) = $[(t_2 – t_1) – (t_4 – t_3)]/2$ = [(t-ms) – (t-sm)]/2
- (propagation time) = $[(t_2 – t_1) + (t_4 – t_3)]/2$ = [(t-ms) + (t-sm)]/2
- Can rewrite the offset as
- Offset = $t_2 – t_1$ – (propagation time) = (t-ms) – (propagation time)
- If the link is not symmetric
  - The propagation time computed as above is the mean of the master-to-slave and slave-to- master propagation times
  - The offset is in error by the difference between the actual master-to-slave and mean propagation times
  - IF you know the asymmetry, the standard specifies how to correct for it.

Ordinary clock:

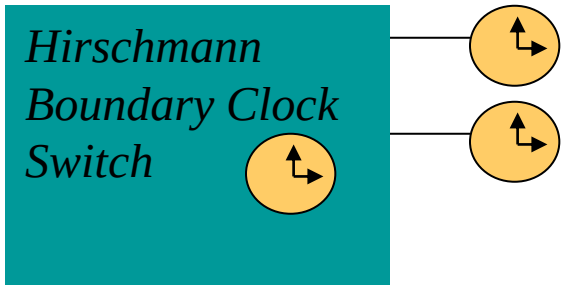


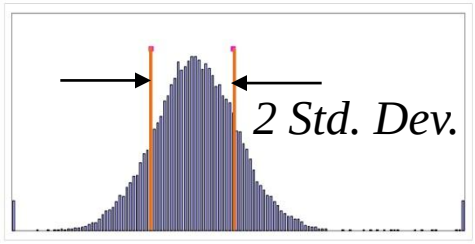


Synchronization accuracy results

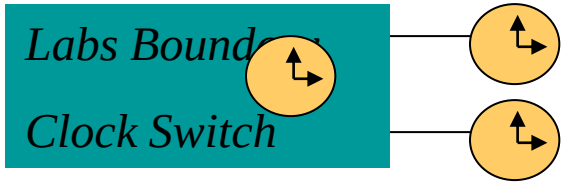


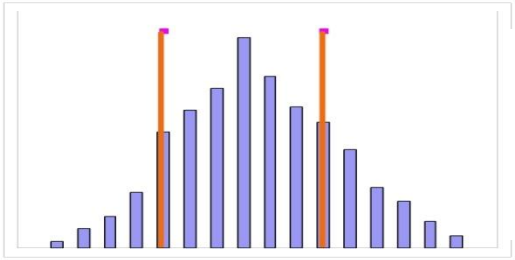
Offset: nanoseconds
Std. Dev: 121 ns
Max/min: -430,+720 ns
RCS switch, 8 hours





Offset: nanoseconds
Std. Dev: 24 ns
Max/min: -100,+100 ns
84 hours



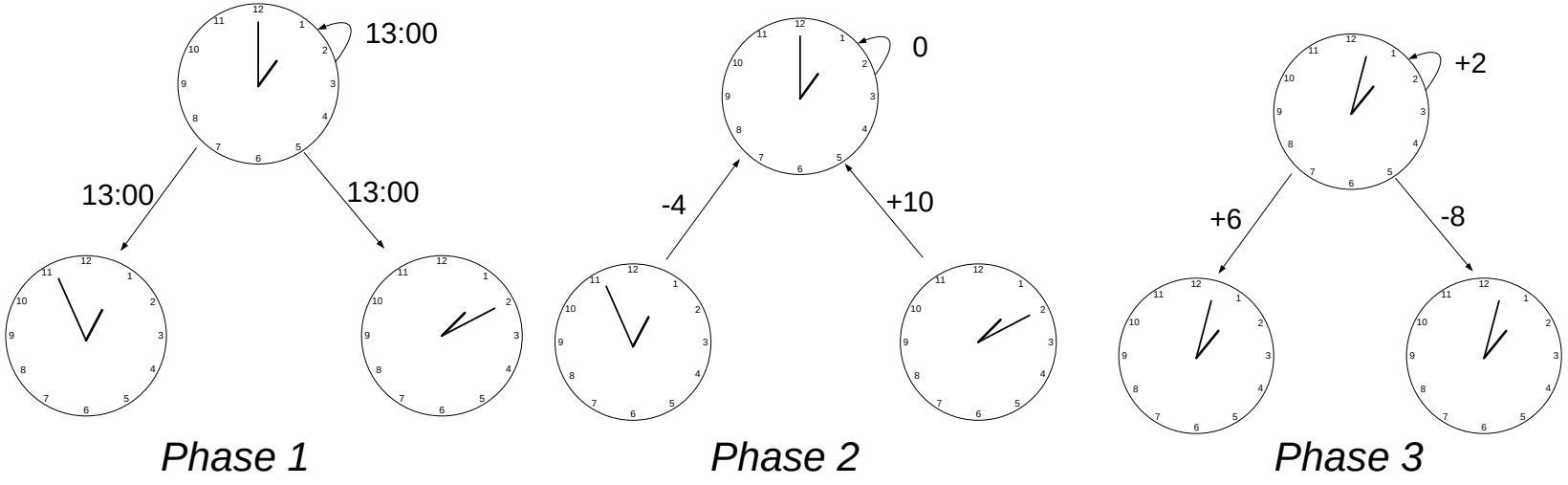


Offset: nanoseconds
Std. Dev: 15 ns
Max/min: -35,+45 ns
1 hour

For all of the above the clock resolution was 20-25 ns

Algorithmus von Berkeley (1989)

- Annahme: kein UTC-Empfänger verfügbar
- Algorithmus (zentral, intern):
  - ein Rechner agiert als aktiver Time-Server.
  - Der Server fragt periodisch die Zeiten/Unterschiede aller anderen Rechner ab (Phase 1) und ermittelt den Durchschnittswert (Phase2).
  - In Phase 3 wird der errechnete Wert an alle anderen Uhren ausgegeben.



NTP: Network Time Protocol (1982)

- Problem: Die angegebenen Algorithmen funktionieren nur in kleinen statischen Netzen.
- Das NTP Protokoll bietet eine Möglichkeit in großen Netzen eine Synchronisation zu gewährleisten.
- Die Netze können dabei dynamisch konfiguriert werden, um eine zuverlässige Synchronisation zu gewährleisten.
- Die Grundstruktur von NTP ist ein hierarchisches Modell (mit verschiedenen Strata/Schichten).
  - Der Dienst wird durch ein verteiltes Serversystem geleistet.
  - Primäre Server sind direkt mit einer UTC-Quelle verbunden.
  - Sekundäre Server synchronisieren sich mit primären Servern usw.
  - Jede zusätzliche Schicht verursacht einen zusätzlichen Zeitversatz von 10-100ms.