

Zeitsynchronisation in Sensornetzen

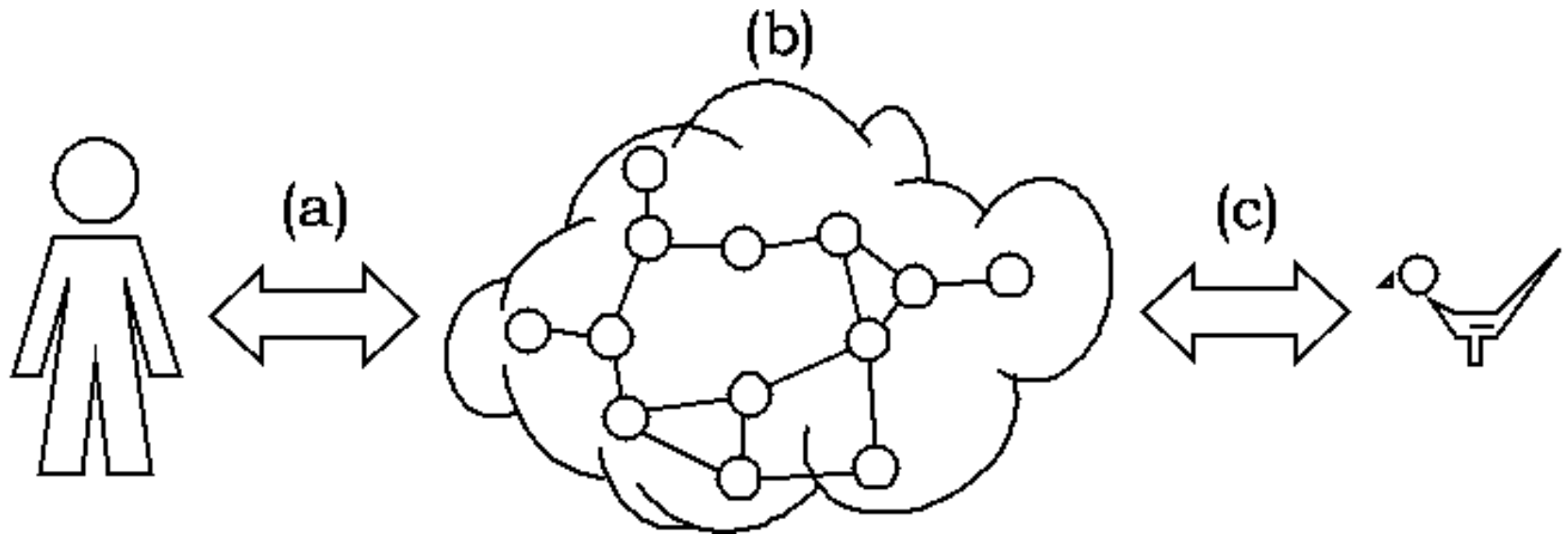
Kay Römer
ETH Zürich
Switzerland

Überblick

- Wozu wird Zeit gebraucht?
- Warum Forschung notwendig?
- Varianten von Synchronisation
- Eigenschaften von Sensorknoten
- Einmalige Synchronisation eines Knotens
- Synchronisation über längere Zeit
- Synchronisation mehrerer Knoten
- Fallbeispiele

1. Wozu wird Zeit gebraucht?

Wozu Zeit?



- (a) Interaktion Nutzer - Sensornetz
- (b) Koordination im Sensornetz
- (c) Interaktion Sensornetz - reale Welt

Nutzer - Sensornetz

- Spezifikation der Beobachtungsaufgabe
 - *Wann* soll etwas beobachtet werden?
- Interpretation der Ergebnisse
 - *Wann* ist ein Ereignis beobachtet worden?
 - *Wie lange* dauerte es?
 - *Wie schnell* war das Objekt?

Sensornetz – Reale Welt

- Zusammenfügen verteilter Beobachtungen
 - Mehrere Sensorknoten kooperieren
 - Bsp. Geschwindigkeit: Wie lange brauchte ein Objekt von einem Knoten zum nächsten?
- Separation mehrerer physischer Ereignisse
 - Welche Beobachtungen gehören zu welchem physischen Ereignis?
 - Zeitlicher Abstand zwischen Ereignissen!
- Konsistente Messungen
 - Knoten müssen gleichzeitig messen, wenn reale Welt sich sehr schnell verändert!

Koordination Sensorknoten

- Drahtlose Kommunikation
 - Exklusiver Zugriff auf Kommunikationskanal
- Energieeffizienz
 - Zeitweise Abschalten des Radios
 - Koordiniertes Aufwachen nötig
- Nachrichtenreihenfolge
 - Bearbeitung von Nachrichten in der Reihenfolge wie sie generiert wurden
- Lokalisierung
 - Signallaufzeit \sim Abstand

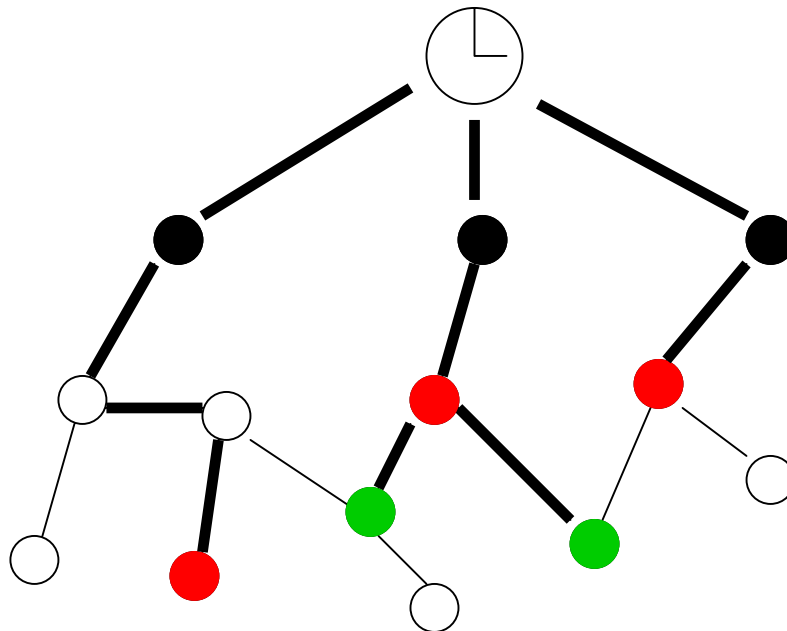
Genauigkeit?

- Hängt stark von der Applikation ab
- Von wenigen Mikrosekunden ...
 - Kommunikation
 - Lokalisation
 - Zusammenfügen von Ereignissen
- ... bis Minuten
 - Aktivierung (von 10 bis 12.30 Uhr ...)

2. Warum Forschung Notwendig?

Warum Neue Lösungen?

- Probleme mit Network Time Protocol (NTP)
 - Infrastruktur
 - Energie
 - Netzdynamik
 - Konfiguration



Stratum 1

Stratum 2

Klienten

Infrastruktur

- Stratum 1 Server haben Zugriff auf eine genaue globale Zeit
 - GPS
 - DCF77
- In Sensornetzen oft nicht möglich oder wünschenswert

Energie

- Server lauschen ständig auf Anfragen von „Kindern“
- In Sensornetzen Energie sehr knappe Ressource
 - Kommunikation dominiert Energieverbrauch
 - Selbst „Lauschen“ kostet sehr viel Energie
 - Vergleichbar mit Datenempfang

Netzdynamik

- NTP basiert auf statischem Netz
 - Feste Pfade zwischen Nachbarn im Overlay-Netz
- In Sensornetzen Netz sehr dynamisch
 - Knoten kommen, gehen, sind mobil, temporäre Störungen
 - Pfade u.U. sehr instabil
 - Kurzzeitige Netz-Partitionierung möglich

Konfiguration

- Statische Konfiguration des Overlay Netzes
 - Wer ist mein Vater?
 - Administrator für überschaubares Netz zuständig
- In Sensornetzen kaum möglich
 - Konfiguration muss an dynamische Veränderungen angepasst werden
 - Ein Administrator hätte u.U. tausende Knoten individuell zu konfigurieren

3. Varianten von Synchronisation

Arten von Synchronisation

- Intern vs. Extern
- Lebenszeit: immer vs. bei Bedarf
- Scope: alle vs. einige
- Rate vs. Offset
- Skalentransformation vs. Uhrensynchron.
- Zeitpunkte vs. Intervalle

Intern vs. Extern

- Intern

- Knoten einigen sich auf eine gemeinsame Zeit, egal welche

- Extern

- Zeit wird von aussen vorgegeben
- Alle Knoten müssen diese Zeit verwenden

Lebenszeit

- Wie lange müssen Uhren synchronisiert bleiben?
 - Immer
 - Nur dann, wenn wirklich benötigt
- Energie!

Scope

- Welche Knoten müssen synchronisiert werden?
 - Alle
 - Nur manche, die es gerade brauchen
- Energie!

Rate vs. Offset

- Raten-Synchronisation
 - Uhren laufen gleich schnell, zeigen aber unterschiedliche Zeiten
 - Ausreichend, wenn Dauer gemessen werden soll
- Offset-Synchronisation
 - Wenn man die Zeit ausliest, zeigen Uhren die gleiche Zeit
 - Notwendig, wenn Zeitpunkte gemessen werden müssen

Skalen vs. Uhren

- Uhrensynchronisation
 - Lokale Uhren werden auf gleiche Zeit eingestellt
- Skalentransformation
 - Uhren sind nicht synchronisiert, definieren unterschiedliche Zeitskalen
 - Zeitstempel werden zwischen Skalen transformiert

Zeitpunkte vs. Intervalle

- Zeitpunkte

- Uhr gibt aus: $\text{Zeit} = 5$
- Suggestiert korrekte Zeit, ist aber fast immer fehlerhaft

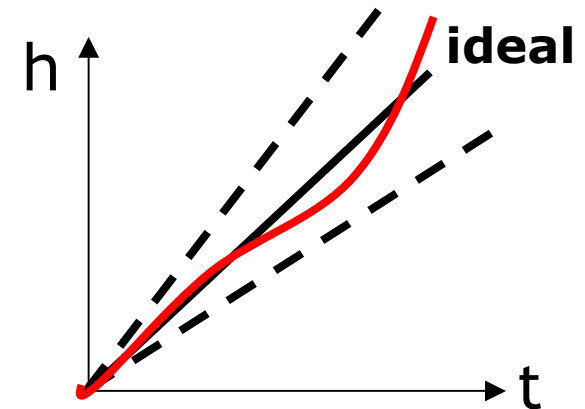
- Intervalle

- Uhr gibt aus: $4 < \text{Zeit} < 5$
- Suggestiert ungenaue Zeit, ist aber immer korrekt

4. Eigenschaften von Sensorknoten

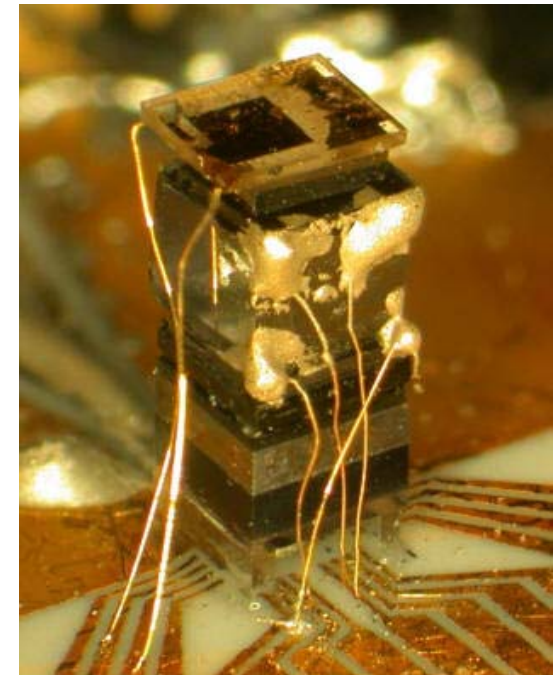
Hardware-Uhren

- Zählen Zeitintervalle fester Länge
- Basieren auf Quarz-Oszillatoren
- Ungenauigkeit (Drift)
 - Wird in ppm (parts per million) gemessen
 - Abweichung von idealer Zeit
 - Typischerweise $< 10\text{-}100\text{ppm}$ in WSN
 - Bsp. Nach 1h Abweichung zwischen $0.04 - 0.4 \text{ s}$



Atomuhr?

- Kleinste Atomuhr der Welt (NIST, Boulder)
 - 5 mm³
 - 30 mA, 2.5 V
 - Genauigkeit: 3.5×10^{-4} ppm
 - 1s Abweichung in 300 Jahren
 - Faktor 10^5 besser!



Kommunikation

- Schwankungen in der Verzögerung entscheidend für Zeitsynchronisation
 - **Sendezeit (+)**
 - Applikation schickt Nachricht ab -> Betriebssystem versucht Nachricht zu übertragen
 - **Zugriffszeit (++)**
 - Wartezeit, bis Kommunikationsmedium frei wird
 - **Signallaufzeit (--)**
 - Ausbreitung Radiowellen
 - **Empfangszeit (-)**
 - Empfang der Daten -> Applikation erhält Nachricht

MAC Timestamping

- Protokoll auf niedrigster Ebene
- Eliminiert weitestgehend
 - Sendezeit
 - Zugriffszeit
 - Empfangszeit

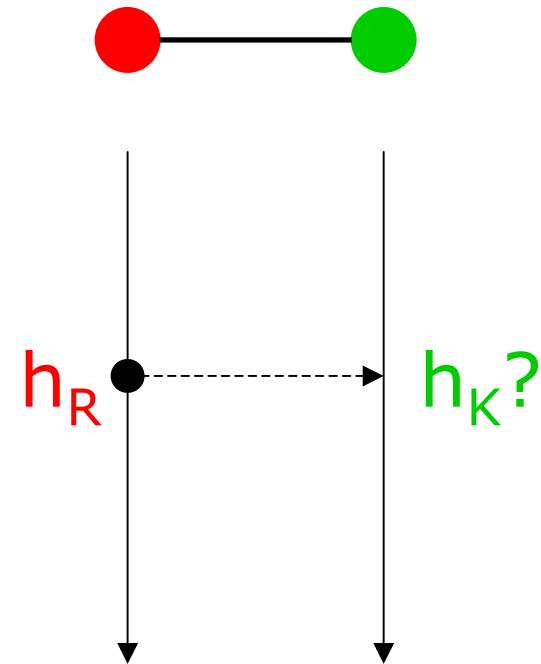
Broadcast

- Mehrere Empfänger nutzen gleiche Nachricht
- Eliminiert weitestgehend
 - **Sendezeit**
 - **Zugriffszeit**

5. Einmalige Synchronisation eines Knotens

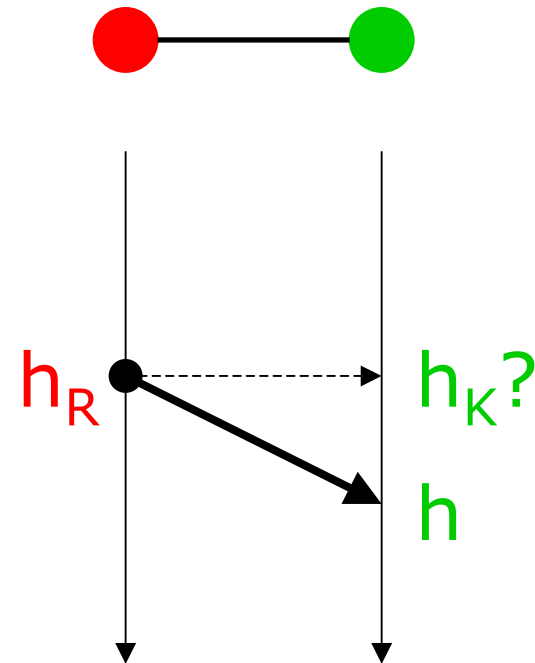
Synchronisation eines Knotens

- **R: Referenz**
- **K: Klient**
- Synchronisations-Problem:
 - Wenn R's Uhr h_R zeigt, wie ist dann K's Zeit h_K ?
- Klient kann dann seine Uhr um $h_R - h_K$ korrigieren



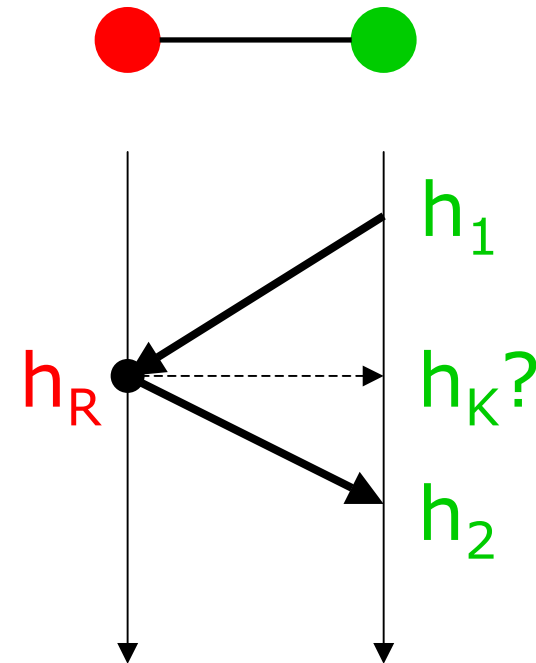
Unidirektional

- Referenz sendet Nachricht, die h_R enthält
- Klient: $h_k := h$
- Geht mit beliebig vielen Klienten (Broadcast!)
- Obere Schranke für h_k



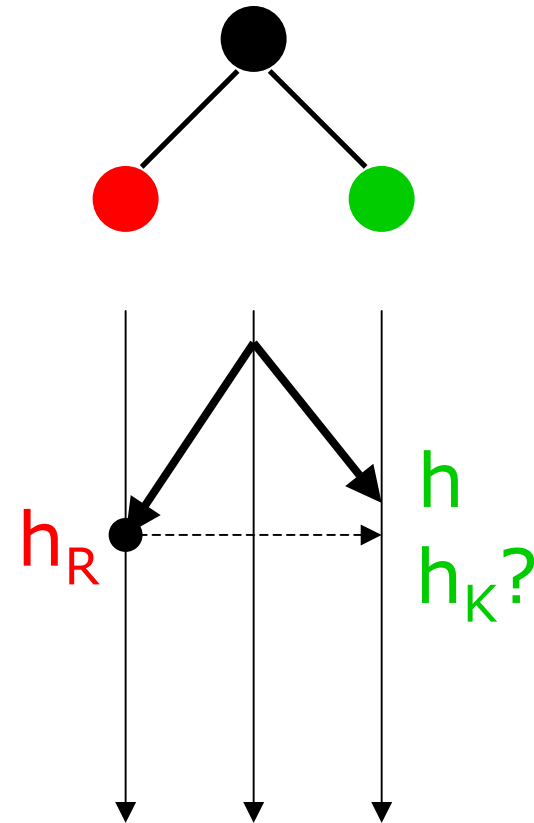
Round Trip

- Klient sendet Nachricht, Referenz antwortet mit h_R
- Klient: $h_k := (h_2 - h_1)/2$
- Genau, wenn Verzögerung konstant
- Aufwendig mit mehreren Klienten
- Obere und untere Schranke für h_k



Referenz Broadcast

- Beacon sendet Broadcast an Referenz und Klient
- Klient: $h_k := h$
- Sehr genau, da Broadcast nahezu gleichzeitig empfangen wird
- Geht mit beliebig vielen Klienten



6. Synchronisation über längere Zeit

Dauerhafte Synchr.

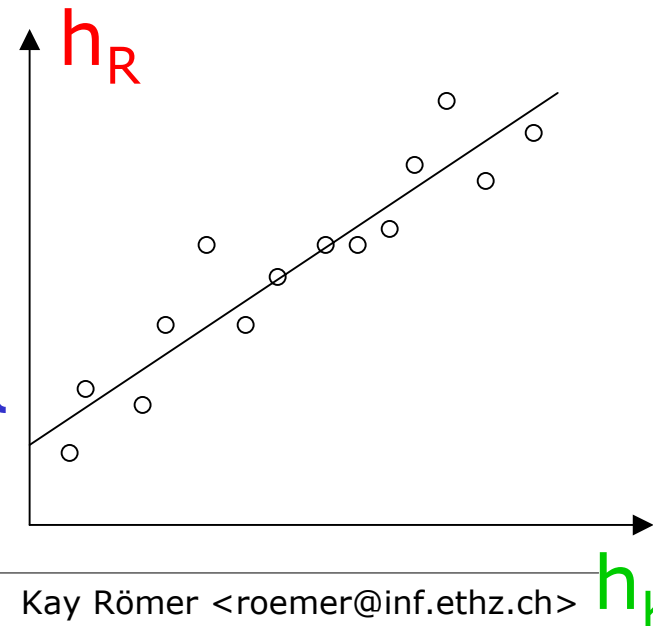
- Einmalige Synchronisierung hält nur begrenzte Zeit wegen Drift!
- Daher:
 - Periodisch wiederholen
 - Ganggeschwindigkeit der Uhren anpassen

Periodische Wiederholung

- Genauigkeit hängt von Häufigkeit der Synchronisierung ab
- Häufigkeit begrenzt durch Dauer des Synchronisationsvorgangs
- Nachteil: hoher Energieaufwand

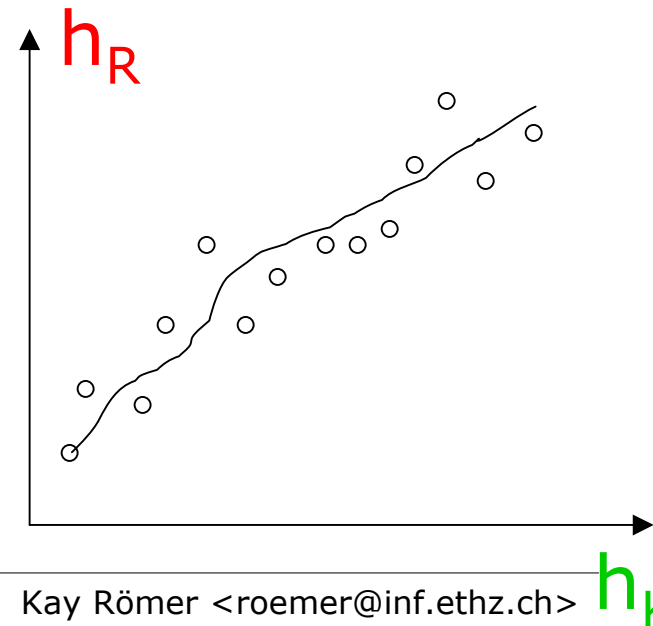
Lineare Regression

- Um wieviel läuft die Uhr des Klienten zu schnell/langsam im Vergleich zur Referenz?
- Mehrere Messungen und linearer Regression
- Steigung der Geraden entspricht Gangunterschied
 - Steigung 1 = gleich schnell
- Nachteil: Speicherbedarf



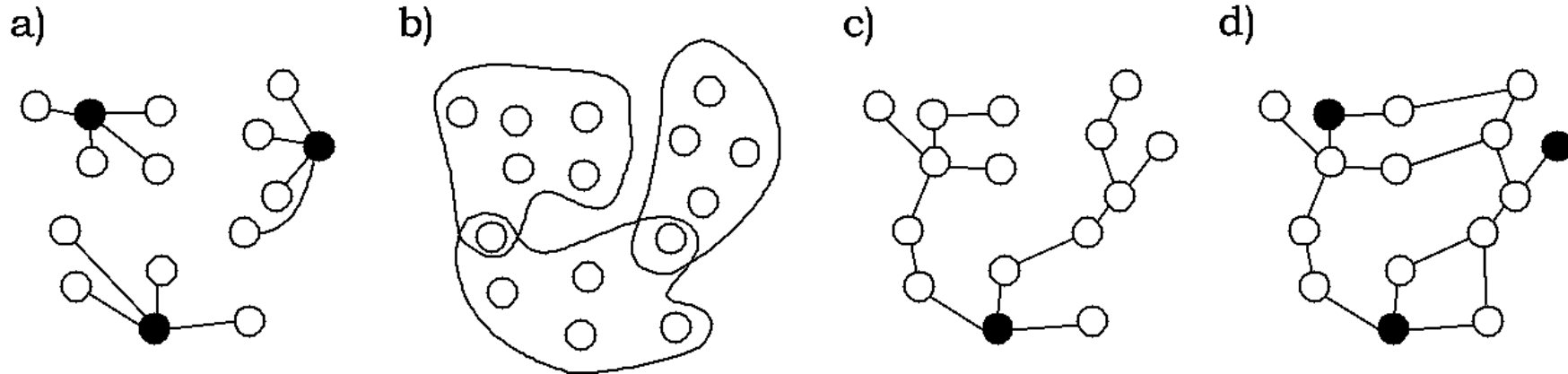
Phase-Locked Loops

- Laufende Abschätzung der aktuellen Zeit
 - Wenn neuer Messwert grösser als aktuelle Abschätzung, dann Rate verringern
 - Wenn neuer Messwert grösser, dann Rate erhöhen
- Nachteil: lange Konvergenzzeit



7. Mehrere Klienten

Mehrere Klienten



- (a) Out-of-band
- (b) Clustering
- (c) Baum
- (d) Unstrukturiert

8. Fallbeispiele

Einige Beispiele

- RBS: Reference Broadcast Synchronization
 - [Jeremy Elson, UCLA, 2002](#)
- TPSN: Timing Sync Protocol for Sensor Networks
 - [Saurabh Ganeriwal, UCLA, 2003](#)
- TSS: Time Stamp Synchronization
 - [Kay Römer, ETHZ, 2001](#)

RBS

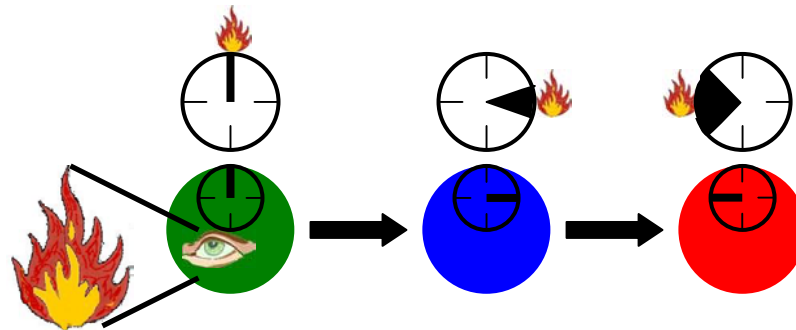
- Clustering
- In jedem Cluster Referenz Broadcast
- Lineare Regression
- Gateways transformieren Zeit zwischen Clustern
- Genauigkeit in einem Cluster auf Berkeley Motes ca. 11 us

TPSN

- Baum
- Kinder synchronisieren sich mit Vätern via Round Trip Messungen
- Genauigkeit auf Berkeley Motes zwischen Nachbarn ca. 17 us

TSS

- Uhren nicht synchronisiert
- Transformation von Zeitstempel entlang eines Nachrichtenpfades
- Round Trip zwischen Nachbarknoten
- Intervalle
- Genauigkeit ca. 1 ms



Zusammenfassung

- Altes Problem, neue Anforderungen, angepasste Lösungen
 - Energie, Keine Infrastruktur, Selbst-Konfiguration, Netzdynamik
 - „Feinde“ der Synchronisation besonders ausgeprägt in WSN

Weitere Informationen

- K. Römer, P. Blum, L. Meier: „Time Synchronization and Calibration in Wireless Sensor Networks“
- Buchkapitel in I. Stojmenovic (Editor), „Wireless Sensor Networks“, Wiley 2005
- Vorabversion auf www.inf.ethz.ch/~roemer