

# Synchronisation in Sensornetzen

## Motivation

In vielen Anwendungen erhöht sich der Nutzen der Messwerte erheblich, wenn die genaue Zeit der Entstehung bekannt ist. Lokale und absolute Zeitmessungen wird auch zum Betrieb des Sensornetzes benötigt, so z. B. zur Lokalisierung oder für die Kommunikation.

## Zeitmessung nötig für Interaktion des Benutzers mit dem Netzwerk

Daten werden nicht lokal ausgewertet, sondern an entfernte Zentrale übermittelt. Wenn dort Daten mit Zeitstempeln versehen sind, können besser Rückschlüsse auf Ereignisse vor Ort getroffen werden (z. B. Geschwindigkeit v. Tierwanderungen, Spechte detektieren, Dauer einer Erschütterung des Bodens, Interpretation von Ereignissen am Tag anders als bei Nacht, lokalisieren v. Ereignissen durch Laufzeitunterschiede bei sehr genauer Zeit möglich).

## Zeitmessung zur Koordination innerhalb des Netzes

Positionsbestimmung über Signallaufzeit bzw. GPS, MAC Zugriff per TDMA (Time Division Multiplexing) d. h. Zeit-Slots, Abschätzen des Energievorrates, wenn Messung nicht möglich mit Vorbereitung auf Take-Over durch andere Knoten, Ermitteln der Roundtrip-time (genauer Takt vs. kleiner Offset).

## Interaktion der Knoten mit der realen Welt

Ampelsteuerung zwischen Uhr 22:00-05:00 zur Flussoptimierung abschalten, Gebäudeüberwachung bei Nacht, Temperaturmessungen zur vollen Stunde, ohne Trigger durch Netzwerk, Tierbeobachtung im Tag- bzw. Nachtmodus.

## Zeiten und Fehlerarten

Reference Broadcast Sync.

RoundTrip Time Sync. (NTP)

Driftberechnung in RBS vs. RTT Sync.

Time Protocol for Sensorn Netw.

Intervallschätzung f. Ereignisse

Vergleiche von Zeitintervallen

# Synchronisation in Sensornetzen

## Motivation

### Systeme und Metriken zur Zeitmessung

- TAI:** Temps Atomique International ist eine gemittelte Zeit aus mehreren Cäsium Atomuhren, die über die ganze Welt verteilt sind. Das Cäsium wechselt seinen elektrischen Zustand 9.192.631.770 mal/Sekunde. Diese Zustandswechsel werden gezählt, um den Ablauf einer Sekunde zu messen. Dabei entsteht eine Abweichung von etwa  $10^{-12}$  pro Uhr, die gemittelte Zeit mehrerer Atomuhren ist entsprechend genauer.
- UTC:** Temps Universel Coordonné ist die weltweit vereinbarte korrigierte Zeit, die sich an die TAI anlehnt, jedoch in manchen Jahren um eine Schaltsekunde korrigiert werden muss, da die Erdumdrehung durch die Gezeiten abgebremst wird, sich der Mond von der Erde entfernt, die Sonne kleiner wird etc.
- NTP:** Network Time Protocol mit dem NTP Zeitstempel, der aus 64 Bit besteht. Dabei stehen die ersten 32 Bit für einen Sekundenzähler, ab dem 1. Januar 0:00 Uhr im Jahr 1900, die letzten 32 Bit bezeichnen den Bruchteil einer Sekunde, d. h. mit einer Genauigkeit von  $1/2^{32}$  Sekunden.

## Zeiten und Fehlerarten

Reference Broadcast Synchronisation.

RoundTrip Time Synchronisation. (NTP)

Driftberechnung in RBS vs. RTT Synchronisation.

Time Protocol for Sensor Networks.

Intervallschätzung für Ereignisse

Vergleiche von Zeitintervallen

# Synchronisation in Sensornetzen

## Motivation

### Naive Lösung für Synchronisation

Anfrage bei einem Knoten, der selbst die genaue Zeit hat und Korrigieren der eigenen Uhr. Ist die zu erwartende Laufzeit der Nachricht klein, vor allem im Verhältnis zur benötigten Genauigkeit der Uhr, so reicht die einfache Nachfrage aus. Sonst kann man die Latenz zwischen Sender der Uhrzeit und Empfänger wie folgt unterteilen:

- Sendezeit:** Die Zeit, die der Sender braucht um die Nachricht zu erstellen, ein Multitasking eine Zeitscheibe zu bekommen, evtl. bestehende Warteschlangen abzubauen etc.
- Zugriffszeit:** Gemeint ist die Zeit die vergeht, bis Zugriff auf das Medium gewährt werden kann (siehe CSMA Verfahren etc.) Insbesondere in Sensornetzen bei allen Verfahren mit Contention-Phase wie S-MAC. T-MAC etc.
- Ausbreitungszeit:** Die Zeit, die die Nachricht selbst auf dem Medium braucht. Zwischen Sendern vernachlässigbar, über das Netzwerk hinweg können sich jed. lange Latenzzeiten akkumulieren, vor allem bei großer Anzahl Hops.
- Empfangszeit:** Analog zur Zeit für das Senden, d. h. gemeint ist die Latanz, bis die Nachricht in der Anwendungsschicht angelangt ist.

## Zeiten und Fehlerarten

Reference Broadcast Synchronisation

RoundTrip Time Synchronisation (NTP)

Driftberechnung in RBS vs. RTT Synchronisation

Time Protocol for Sensor Networks

Intervallschätzung für Ereignisse

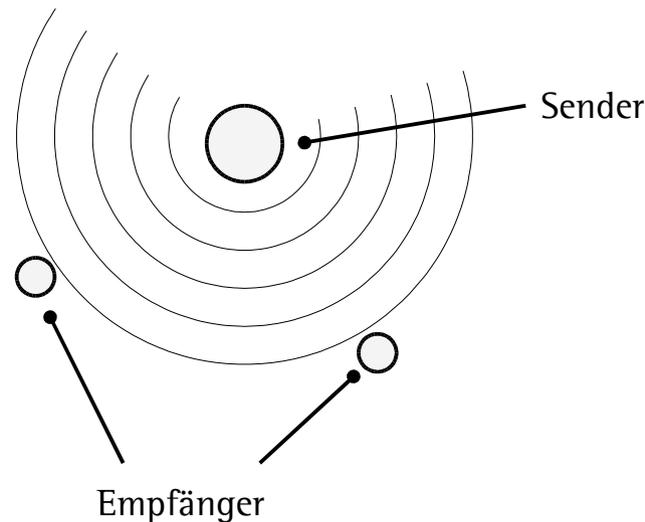
Vergleiche von Zeitintervallen

# Synchronisation in Sensornetzen

## Motivation

### Lösung für Sensornetze: Reference-Broadcast Synchronization

**Idee:** Ausnutzung des Broadcast-Kanals, in dem sich Nachrichten für alle gleichschnell ausbreiten und ankommen (abgesehen von verschieden weit entfernten Knoten). Ein Sender sendet ein bel. Signal (evtl. ohne speziellen Inhalt) an alle Empfänger.



In der Regel kann die Ausbreitungsgeschwindigkeit des Funksignals für kurze Strecken vernachlässigt werden. Als Quelle der Varianz für die Latenz bleibt die Empfangszeit in den unterschiedlichen Knoten selbst. Sende- und Kanal-Zugriffszeit beim Sender sind dagegen für alle Empfänger ein und desselben Broadcast-Signales gleich.

Zeiten und Fehlerarten

Reference Broadcast Synchron.

RoundTrip Time Sync. (NTP)

Driftberechnung in RBS vs. RTT Sync.

Time Protocol for Sensor Netw.

Intervallschätzung f. Ereignisse

Vergleiche von Zeitintervallen

# Synchronisation in Sensornetzen

## Motivation

### Lösung für Sensornetze: Reference-Broadcast Synchronization

Jeder Empfänger merkt sich seine lokale Zeit, zu der er das Signal empfangen hat. Bis jetzt ist damit nicht viel gewonnen. Ledoch kann so beim nächsten Austausch von Informationen zwischen beliebigen Knoten die Drift der Uhren (zum Zeitpunkt des Empfangs des Signals) ermittelt werden.

**Beispiel:** A empfäng um 17:05 ein Broadcast-Signal. Später kommunizieren A und B miteinander. B fügt dem Paket gleich noch die Information hinzu, dass es zuletzt ein Broadcast-Signal um 17:03 empfangen hat. Glaubt A der Zeit von B mehr als seiner eigenen Uhr, so kann er diese um 2 Minuten zurückstellen. Zusätzliche Nachrichten wurden so nicht ausgetauscht. Trotzdem konnte A seine Uhr stellen.

**Vorteil:** Als Broadcast-Signal können auch Nachrichten herangezogen werden, die eigentlich eine ganz andere Funktion haben, also z. B. das RTS oder CTS in Funknetzen (siehe Kapitel MAC-Layer).

**Nachteil:** Alle Empfänger müssen sich in der gleichen „Empfangsdomäne“ befinden.

[Ausgleichen der Screw?]

Zeiten und Fehlerarten

Reference Broadcast Synchron.

RoundTrip Time Sync. (NTP)

Driftberechnung in RBS vs. RTT Sync.

Time Protocol for Sensorn Netw.

Intervallschätzung f. Ereignisse

Vergleiche von Zeitintervallen

# Synchronisation in Sensornetzen

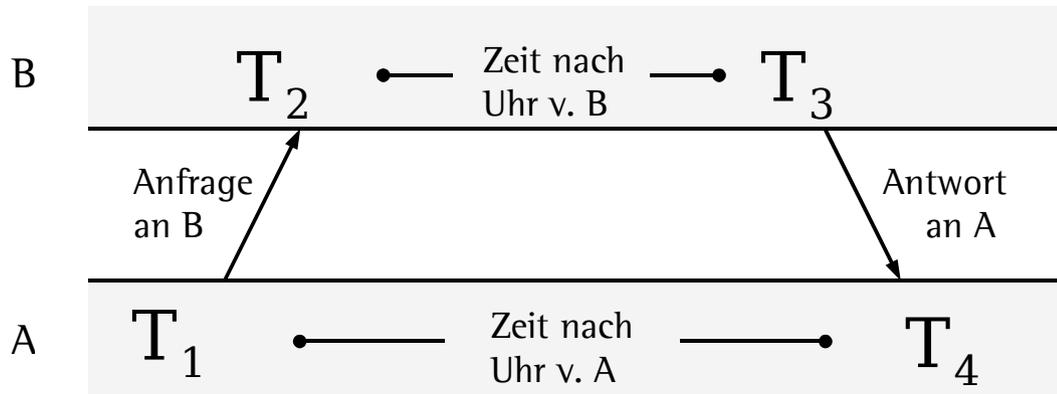
Das Network Time Protocol (NTP)

Abschätzung für reine Paketlaufzeit Delta und Uhrendrift Theta

$$\delta = \frac{(T_4 - T_1) - (T_3 - T_2)}{2}$$

$$\theta = \frac{(T_2 - T_1) + (T_3 - T_4)}{2}$$

theta > 0 bedeutet, dass die Uhr von B vor geht, im Vergleich zur Uhr von A



Zeiten und Fehlerarten

Reference Broadcast Sync.

RoundTrip Time Sync. (NTP)

Driftberechnung in RBS vs. RTT Sync.

Time Protocol for Sensornetw.

Intervallschätzung f. Ereignisse

Vergleiche von Zeitintervallen

## Synchronisation in Sensornetzen

### Das Network Time Protocol (NTP)

Abschätzung für Roundtriptime Delta und Uhrendrift Theta

$$\delta = \frac{(T_4 - T_1) - (T_3 - T_2)}{2} \quad \theta = \frac{(T_2 - T_1) + (T_3 - T_4)}{2}$$

Probe: Die Uhren seien um  $e$  Zeiteinheiten verschoben, die einfache Paketlaufzeit betrage  $d$  ZE.

$$T_2 = T_1 + d + e$$

$$T_4 = T_3 + d - e$$

$$\delta = \frac{(T_3 + d - e - T_1) - (T_3 - (T_1 + d + e))}{2} = d$$

$$\theta = \frac{(T_1 + d + e - T_1) + (T_3 - (T_3 + d - e))}{2} = e$$

Zeiten und Fehlerarten

Reference Broadcast Synch.

RoundTrip Time Sync. (NTP)

Driftberechnung in RBS vs. RTT Sync.

Time Protocol for Sensornetw.

Intervallschätzung f. Ereignisse

Vergleiche von Zeitintervallen

# Synchronisation in Sensornetzen

## Fehleranalyse der Drift-Berechnung zwischen RTT und RBS

RTT basierte Verfahren: Die RoundTrip Time eines Nachrichtenaustausches in beide Richtungen wird herangezogen, um die wahre Drift der Uhren zu berechnen.

RBS basierte Verfahren: Bei der Reference Broadcast Synchronization werden eine Menge von Empfängern durch einen zentralen Signalgeber synchronisiert.

$D_K$  = Drift zwischen Knoten K und der Realzeit

$S_K$  = Sendezeit, d. h. Zeit um Warteschlange abzubauen, auf Zuteilung einer Zeitscheibe für Prozessor warten, CRC-Polynom berechnen usw., jedoch noch nicht Zugriff auf Kanal

$A_K$  = (Access) Wartezeit für Zugriff auf den Kanal, besonders bei MAC-Verfahren mit Contention-Phase (d. h. Konkurrenz um den gemeinsamen Kanal)

$P_{K,L}$  = (Propagation) Signalausbreitung zwischen Knoten K und L

$R_L$  = (Receive) Zeit zum Empfangen des Paketes und evtl. Durchreichen durch die Schichten des Netzwerkmodells bis zur Anwendungsebene in Knoten L

$T_1$  = Durch einen Knoten angenommene Zeit mit Index 1

$t_1$  = die zu  $T_1$  passende tatsächliche Realzeit (Drift ist also  $T_1 - t_1$ )

Zeiten und Fehlerarten

Reference Broadcast Sync.

RoundTrip Time Sync. (NTP)

Driftberechnung in RBS vs. RTT Sync.

Time Protocol for Sensornetw.

Intervallschätzung f. Ereignisse

Vergleiche von Zeitintervallen

# Synchronisation in Sensornetzen

## Fehleranalyse der Drift-Berechnung zwischen RTT und RBS

Fehler in Drift bei RTT:

$$T_1 = t_1 + D_A$$

$$T_3 = t_3 + D_A$$

$$T_2 = T_1 + S_A + A_A + P_{A,B} + R_B$$

$$T_4 = T_3 + S_B + A_B + P_{B,A} + R_A$$

Drift: 
$$\theta_1 = \frac{(T_2 - T_1) + (T_3 - T_4)}{2}$$

**Fehler in Drift:** Für die geschätzten Zeiten T werden die Realzeiten t eingesetzt, jeweils erweitert um die Fehler, die durch die Kommunikation hinzugefügt werden. Da die Realzeit „nicht von sich selbst abweicht“ (also Drift=0), muss am Ende der reine Netzwerk-Fehler verbleiben.

$$e_{\theta_1} = \frac{(t_1 + D_A + S_A + A_A + P_{A,B} + R_B - (t_1 + D_A)) + (t_3 + D_A - (t_3 + D_A + S_B + A_B + P_{B,A} + R_A))}{2}$$

$$e_{\theta_1} = \frac{S_A + A_A + P_{A,B} + R_B - S_B - A_B - P_{B,A} - R_A}{2}$$

Man kann annehmen, dass die Ausbreitung des Signales von A nach B  $P_{A,B}$  etwa so schnell ist wie die Ausbreitung in die entgegengesetzte Richtung, so dass näherungsweise gilt  $P_{A,B} = P_{B,A}$ . Damit fallen die beiden P-Variablen noch heraus.

$$e_{\theta_1} = \frac{S_A + A_A + R_B - S_B - A_B - R_A}{2}$$

Wie stark sich die restlichen Variablen gegenseitig ausgleichen hängt allein von deren Varianz ab (denn bei Knoten gleicher Bauart ist der Erwartungswert z. B. von  $E(S_A)$  etwa gleich  $E(S_B)$ ). Je kleiner die Varianz ist, desto genauer kann die Drift geschätzt werden.

Zeiten und Fehlerarten

Reference Broadcast Sync.

RoundTrip Time Sync. (NTP)

**Driftberechnung in RBS vs. RTT Sync.**

Time Protocol for Sensorn Netw.

Intervallschätzung f. Ereignisse

Vergleiche von Zeitintervallen

# Synchronisation in Sensornetzen

## Fehleranalyse der Drift-Berechnung zwischen RTT und RBS

Fehler in Drift bei RBS:

$$T_1 = t_1 + D_S$$

$$T_2 = T_1 + S_S + A_S + P_{S,A} + R_A$$

$$T_3 = T_1 + S_S + A_S + P_{S,B} + R_B$$

Drift:

$$\theta_2 = T_2 - T_3$$

Hier sendet ein Knoten S zum Zeitpunkt  $T_1$  ein Signal aus, welches von Knoten A nach dessen Uhr zum Zeitpunkt  $T_2$ , von Knoten B nach Meinung von B zum Zeitpunkt  $T_3$  empfangen wird.

Fehler in Drift:

$$e_{\theta_2} = T_2 - T_3 = t_1 + D_S + S_S + A_S + P_{S,A} + R_A - (t_1 + D_S + S_S + A_S + P_{S,B} + R_B)$$

$$e_{\theta_2} = P_{S,A} + R_A - P_{S,B} - R_B$$

Zum Vergleich:

$$e_{\theta_1} = \frac{S_A + A_A + R_B - S_B - A_B - R_A}{2}$$

Die Latenzzeiten beim Sender heben sich bei RBS heraus. Dafür ist die Signallaufzeit zwischen Sender und unterschiedlichen Empfängern von der Entfernung abhängig und verbleibt so im Rest. Außerdem wird der Term nicht halbiert. Bei kleiner Varianz gleichen sich jedoch die R-Zeiten etwa aus, und der P-Anteil ist bei Funknetzen sehr klein.

Zeiten und Fehlerarten

Reference Broadcast Sync.

RoundTrip Time Sync. (NTP)

Driftberechnung in RBS vs. RTT Sync.

Time Protocol for Sensornetw.

Intervallschätzung f. Ereignisse

Vergleiche von Zeitintervallen

# Synchronisation in Sensornetzen

## TPSN: Timing-sync Protocol for Sensor Networks

(Aus der gleichnamigen Veröffentlichung von S. Ganeriwal, R. Kumar, M. B. Srivastava)

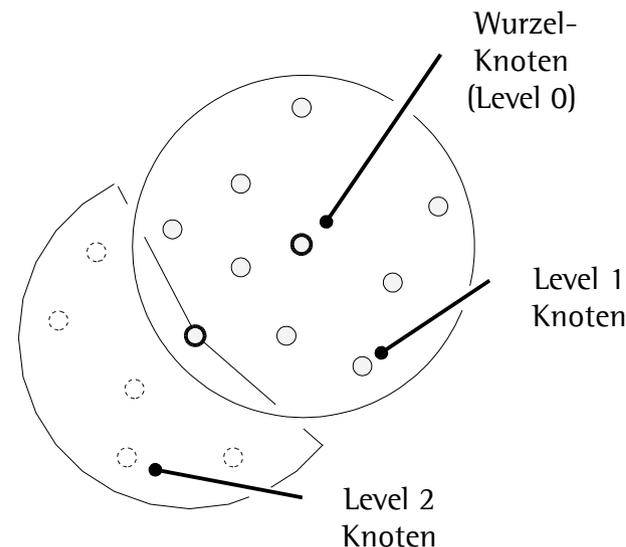
### Initialisierungsphase

Die Initiative von TPSN geht immer vom geringeren Level, anfangs vom Wurzel-Knoten aus. Dieser sendet ein `level_discovery` Paket, mit seinem eigenen Level.

Alle Knoten in der Sendereichweite empfangen das Paket und setzen ihr Level auf das um 1 inkrementierte Level aus dem Paket. Dann zieht jeder Knoten einen Zufallstimer auf.

Nach dessen Ablauf sendet auch er ein `level_discovery` Paket. Knoten die a) ihr Level bereits festgesetzt haben ignorieren dieses Paket die anderen Knoten b) setzen ihr Level wieder auf das um 1 erhöhte Level aus dem Paket. Nach einiger Zeit werden auch sie ein eigenes `level_discovery` Paket aussenden etc. Hat am Ende des Flutungsprozesses jeder Knoten ein Level? -> Nein, denn durch die Zufallstimer können Kollisionen auftreten.

Lösung: Ein Knoten N der länger kein gültiges `level_discovery` Paket erhalten hat ergreift die Initiative und sendet ein `level_request` paket. Auf dieses antworten alle Knoten in der Umgebung. N sucht aus allen empfangenen Paketen das mit der kleinsten ID für sich heraus und inkrementiert es um eins.



Zeiten und Fehlerarten

Reference Broadcast Sync.

RoundTrip Time Sync. (NTP)

Driftberechnung in RBS vs. RTT Sync.

Time Protocol for Sensor Netw.

Intervallschätzung f. Ereignisse

Vergleiche von Zeitintervallen

# Synchronisation in Sensornetzen

## Motivation

Lösung für Sensornetze: Time-Sync Protocol for sensor Networks (TPSN)

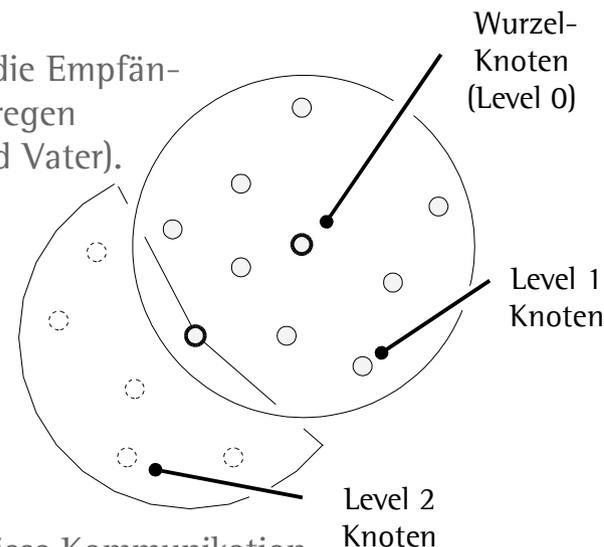
## Zeitsynchronisations-Phase

Der Wurzelknoten erzeugt ein `time_sync` Paket, welches die Empfänger auf Level 1 zur Nachfrage nach der aktuellen Zeit anregen soll (und zwar immer nur paarweise zwischen Knoten und Vater).

Die Empfänger ziehen wie üblich ihren Zufallstimer auf und senden nach dessen Ablauf einen sog. `synchronization pulse` an den Wurzel- bzw. den Vaterknoten. Kommt dieser korrekt an, so sendet der übergeordnete Knoten ein `ACK`, aus dem in der Art von NTP die Roundtripzeit ermittelt werden kann.

Alle Knoten mit höherem Level (hier z. B. 2) überhören diese Kommunikation vorerst, nehmen jedoch den `synchronisation pulse` aus Level 1 zum Anlass, einen eigenen Timer aufzuziehen. Dieser wird so gewählt, dass der aktive Level 1 Knoten seine Synchronisation abschließen kann. Nach einiger Zeit laufen die ersten Timer der Level 2 Knoten ab, so dass diese ihrerseits `synchronization pulses` an ihre Level 1 Väter aussenden usw.

Der `synchronisation pulse` hat den Nebeneffekt, dass ein Vaterknoten erkennen kann, dass er solche nie erhält. Evtl. wissen die Kinder (aufgrund von Kollisionen) nichts von der aktuellen Phase. Dann kann er erneut ein triggerndes `time_sync` Paket verschicken.



Zeiten und Fehlerarten

Reference Broadcast Sync.

RoundTrip Time Sync. (NTP)

Driftberechnung in RBS vs. RTT Sync.

Time Protocol for Sensor Netw.

Intervallschätzung f. Ereignisse

Vergleiche von Zeitintervallen

# Synchronisation in Sensornetzen

## Schätzung von Zeit-Intervallen in Sensornetzen

Voraussetzung genauer Uhrzeit ist in Sensornetzen oft unrealistisch weil...

- keine Synchronisationsquelle verfügbar ist
- Uhren aufgrund starker Temperaturschwankungen ungenau sind
- sich Ungenauigkeiten über viele Hops hinweg akkumulieren
- Uhren grundsätzlich unterschiedlich schnell laufen können

Ein Ereignis wird im zeitlichen Bezugssystem des Beobachters detektiert. Anstatt sich einer genauen globalen Zeit anzupassen akzeptiert man die Ungenauigkeit lokaler Zeitmessungen. Jedoch wird bei der Weitergabe einer „Entdeckung“ an einen anderen Sensorknoten der Zeitpunkt der Messung an das lokale Bezugssystem des Empfängers angepaßt.

**Beispiel:** Ein Beobachter A detektiert ein Ereignis um 19:00 und erzählt es gleich darauf B. B's Uhr zeigt 18:00 Uhr an. B weiß also vom gleichen Ereignis zum etwa gleichen Zeitpunkt, nennt diesen Zeitpunkt aufgrund seiner Uhr jedoch 18:00 Uhr. Die Nachricht wird an C weiter erzählt, dessen Uhr wieder 19:00 Uhr anzeigt. Die Tatsache, dass B den Zeitpunkt dazwischen (für sich) als 18:00 Uhr bezeichnet hat, macht nichts.

**Problem:** Zwischen dem Weiterleiten der Nachricht vergeht in Wirklichkeit Zeit, um die die jeweilige Benachrichtigung korrigiert werden muss. Diese entspricht in Computernetzwerken der Zeit, die ein Paket unterwegs ist. Diese Zeit kann durch die halbe Roundtrip Time zwischen zwei Teilnehmern im Netz nur geschätzt werden. Die ganze Roundtrip Time stellt eine Obergrenze für die Weiterleitung einer Nachricht dar. Eine weitere Ungenauigkeit besteht in den unterschiedlich genauen Uhren der Knoten.

Der Ungenauigkeit bei der Messung von Zeitpunkten läßt sich dadurch begegnen, dass ein Intervall abgeschätzt wird, zwischen dessen Unter- und Obergrenze ein Ereignis stattgefunden haben muss.

Zeiten und Fehlerarten

Reference Broadcast Synch.

RoundTrip Time Sync. (NTP)

Driftberechnung in RBS vs. RTT Sync.

Time Protocol for Sensorn Netw.

Intervallschätzung f. Ereignisse

Vergleiche von Zeitintervallen

## Synchronisation in Sensornetzen

### Schätzung von Zeit-Intervallen in Sensornetzen

Verfahren der Schätzung eines Zeit-Intervalles für ein Ereignis nach „Time Synchronization in Ad Hoc Networks“ von Kay Römer

Knoten 1:

$$[r_1, r_1] = [T_{\text{Messung}}, T_{\text{Messung}}]$$

Knoten 2:

$$\left[ r_2 - (s_1 - r_1) \frac{1+p_2}{1-p_1} - \left( r_{tt_2} - \text{idle}_1 \frac{1-p_2}{1+p_1} \right), r_2 - (s_1 - r_1) \frac{1-p_2}{1+p_1} \right]$$

Knoten 3:

$$\left[ r_3 - (s_1 - r_1) \frac{1+p_3}{1-p_1} - (s_2 - r_2) \frac{1+p_3}{1-p_2} - \left( r_{tt_2} \frac{1+p_3}{1-p_2} - \text{idle}_1 \frac{1-p_3}{1+p_1} \right) - \left( r_{tt_3} - \text{idle}_2 \frac{1-p_3}{1+p_2} \right), \dots \right. \\ \left. \dots r_3 - (s_1 - r_1) \frac{1-p_3}{1+p_1} - (s_2 - r_2) \frac{1-p_3}{1+p_2} \right]$$

Knoten N:

$$\left[ r_N - (1+p_N) \sum_{i=1}^{N-1} \frac{s_i - r_i}{1-p_i} - (1+p_N) \sum_{i=1}^{N-1} \frac{r_{tt_i}}{1-p_i} + (1-p_N) \sum_{i=2}^N \frac{\text{idle}_i}{1+p_{i-1}} + r_{tt_N}, r_N - (1-p_N) \sum_{i=1}^{N-1} \frac{s_i - r_i}{1+p_i} \right]$$

Außerplanmäßig:  $r_{tt_3}$  muss nicht mehr transformiert werden. Da es von Knoten 3 selbst gemessen wurde ist es natürlich schon im zeitlichen Bezugssystem von Knoten 3.

Zeiten und Fehlerarten

Reference Broadcast Sync.

RoundTrip Time Sync. (NTP)

Driftberechnung in RBS vs. RTT Sync.

Time Protocol for Sensornetw.

Intervallschätzung f. Ereignisse

Vergleiche von Zeitintervallen

## Synchronisation in Sensornetzen

### Schätzung von Zeit-Intervallen in Sensornetzen

Knoten 2:

$$\left[ r_2 - (s_1 - r_1) \frac{1+p_2}{1-p_1} - \left( rtt_2 - idle_1 \frac{1-p_2}{1+p_1} \right), r_2 - (s_1 - r_1) \frac{1-p_2}{1+p_1} \right]$$

Im Unterschied zur Untergrenze des Intervalls nimmt man hier eine Paketlaufzeit von 0 an <sup>\*1</sup>. Für die Wartezeit  $(s_1 - r_1)$ , die die Nachricht über ein Ereignis in Knoten 1 verweilt nimmt man eine schnelle Uhr in Knoten 1 und eine langsame in Knoten 2 an. Damit wird die Wartezeit gewissermaßen bis zur äußersten noch vertretbaren Grenze minimiert.

Siehe hierzu die separate Erläuterung auf der übernächsten Seite.  $(rtt_2 - idle_1 \times \text{Bruch})$  entspricht der Paketlaufzeit auf dem Hin- und Rückweg zwischen Knoten 1 und 2 und stellt damit eine Obergrenze für den einfachen Nachrichtenaustausch dar (der Erwartungswert ist eigentlich Paketlaufzeit/2). Die Nenner sind jew. so gewählt, dass eine max. Laufzeit errechnet wird. Denn die Untergrenze des Intervalls soll möglichst klein werden. Dazu muss die Uhr von Knoten 1 so schnell und die von Knoten 2 so langsam sein wie es die Ungenauigkeit  $p$  der Uhren gerade noch erlaubt.

Die Wartezeit, die die Nachricht bei Knoten 1 verweilt hat ist bezogen auf dessen Uhr, mit ihrer Ungenauigkeit  $\pm p_1$ . Im langsamsten Fall ist die Uhr von Knoten 1 um den Faktor  $(1-p_1)$  zu langsam und die von Knoten 2 um den Faktor  $(1+p_1)$  zu schnell. Der Quotient ist die längstmögliche Wartezeit bei Knoten 1, bezogen auf das Zeitsystem von Knoten 2.

Mit der Nachricht hat Knoten 1 auch mitgeteilt, dass er selbst seit Zeitpunkt  $r_1$  Kenntnis vom Ereignis hat und dieses zum Zeitpunkt  $s_1$  an Knoten 2 weiterleitete.  $r_1$  und  $s_1$  beziehen sich auf die (ungenau?) Uhr von Knoten 1. Man könnte sagen, dass Knoten 1 die Zeit  $(s_1 - r_1)$  vertrödelt hat, bis er die Nachricht weitergab.

Im Zeitpunkt  $r_2$  erfährt Knoten 2 von einem Ereignis. Damit ist  $r_2$  eine erste Schätzung für den Zeitpunkt des Ereignisses selbst. Auf jeden Fall kann es nicht später als  $r_2$  stattgefunden haben.

<sup>\*1</sup> Warum darf man die Paketlaufzeit einfach weglassen? Antwort: Weil immer nur die gesamte Laufzeit des Pakets auf dem Hin- und Rückweg messbar ist. Es könnte also sein, dass fast die gesamte Laufzeit auf dem Hinweg und kaum Zeit auf dem Rückweg verbraucht wurde. Daher ist die Untergrenze für den Rückweg gerade Null.

Zeiten und Fehlerarten

Reference Broadcast Sync.

RoundTrip Time Sync. (NTP)

Driftberechnung in RBS vs. RTT Sync.

Time Protocol for Sensornetw.

Intervallschätzung f. Ereignisse

Vergleiche von Zeitintervallen

# Synchronisation in Sensornetzen

## Schätzung von Zeit-Intervallen in Sensornetzen

Was hat es mit der Rechnung ( $r_{tt}$ -idle) auf sich?

Hier möchte der Empfänger E der Nachricht gerade eine Obergrenze für die Paket-Laufzeit zwischen ihm und dem Sender S abschätzen. Hierzu könnte er wie auch bei NTP üblich ein Ping zum Sender schicken und auf das Pong warten. Jedoch würde man dadurch zwei zusätzliche Nachrichten erzeugen, was insbes. in Sensornetzen wg. der Energieeffizienz unerwünscht ist.

Lösung: Meist hat E den Sender S bereits früher (aus ganz anderen Gründen als dem der Zeitmessung) schon einmal eine Nachricht geschickt und merkt sich diesen Zeitpunkt  $t_1$ .

Dies ist gewissermaßen bereits das Ping in NTP. S informiert E nun über ein Ereignis und natürlich kennt E den Zeitpunkt  $t_2$  des Eintreffens der Nachricht.

Eine, wenn auch extrem grobe Abschätzung für die Roundtrip Time  $r_{tt}$  wäre nun  $t_2 - t_1$ . Natürlich könnte zwischen der ersten Nachricht von E nach S und der späteren Nachricht von S an E eine lange Zeit vergangen sein. Diese Wartezeit  $idle_1$  (zwischen dem letzten Nachrichteneingang bei S und der letzten Nachricht von S an E) läßt sich von S aber problemlos messen und wird dem Paket mit dem Ereignis einfach hinzugefügt. Das Ereignis ist also gleichzeitig das Pong, so dass sich für die Roundtrip Time nun eine recht gute Abschätzung  $t_2 - t_1 - idle_1$  ergibt.

**Nachteil:** S und E müssen Buch führen über die Zeitpunkte der letzten Nachrichteneingänge ihrer Nachbarknoten

Zeiten und Fehlerarten

Reference Broadcast Sync.

RoundTrip Time Sync. (NTP)

Driftberechnung in RBS vs. RTT Sync.

Time Protocol for Sensor Netw.

Intervallschätzung f. Ereignisse

Vergleiche von Zeitintervallen

## Synchronisation in Sensornetzen

### Schätzung von Zeit-Intervallen in Sensornetzen

Vergleichende Aussagen über Ereignis-Intervalle

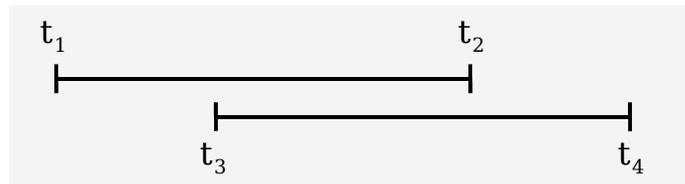
Ebenso wie man Aussagen über die zeitliche Reihenfolge von Ereignissen machen möchte, möchte man dies auch mit den Intervallen tun.

Ereignis  $E_1$  liege im Intervall  $[t_1, t_2]$ , Ereignis  $E_2$  im Intervall  $[t_3, t_4]$ .

Liegt  $E_1$  vor  $E_2$ ?

Ja	wenn $t_2 < t_3$
Nein	wenn $t_1 > t_4$
Vielleicht	sonst

Nähere Analyse von „Vielleicht“: Mit welcher Wahrscheinlichkeit liegt  $E_1$  vor  $E_2$ ?



Liegt  $E_1$  in  $[t_1, t_3]$ , so trat  $E_1$  vor  $E_2$  auf. Dies geschieht mit einer Wahrscheinlichkeit von

$$W_1 = \frac{t_3 - t_1}{t_2 - t_1}$$

Zeiten und Fehlerarten

Reference Broadcast Sync.

RoundTrip Time Sync. (NTP)

Driftberechnung in RBS vs. RTT Sync.

Time Protocol for Sensor Netw.

Intervallschätzung f. Ereignisse

**Vergleiche von Zeitintervallen**

# Synchronisation in Sensornetzen

## Schätzung von Zeit-Intervallen in Sensornetzen

Vergleichende Aussagen über Ereignis-Intervalle

Im anderen Fall muss  $E_1$  nach  $t_3$  aufgetreten sein. Dies geschieht mit einer Wahrscheinlichkeit von

$$\frac{t_2 - t_3}{t_2 - t_1} = (1 - W_1)$$

Damit  $E_1$  grundsätzlich die Chance hat vor  $E_2$  zu erfolgen, muss  $E_2$  selbst im Intervall  $[t_3, t_2]$  liegen. Dies trifft mit einer Wahrscheinlichkeit von

$$\frac{t_2 - t_3}{t_4 - t_3}$$

zu. Folglich liegen  $E_1$  und  $E_2$  mit der folgenden Wahrscheinlichkeit im Intervall  $[t_3, t_2]$ :

$$(1 - W_1) \times \left( \frac{t_2 - t_3}{t_4 - t_3} \right)$$

Zusätzlich muss noch gelten, dass  $E_1$  vor  $E_2$  stattfindet. Wird jeder Zeitpunkt im Intervall gleich wahrscheinlich angenommen, so tritt dies in 50% der Fälle auf, in denen beide Ereignisse im gemeinsamen Intervall liegen, also mit

$$W_2 = \frac{1}{2} \times (1 - W_1) \times \left( \frac{t_2 - t_3}{t_4 - t_3} \right)$$

$W_1 + W_2$  gibt dann die Wahrscheinlichkeit für den Fall der Fragestellung an.

Zeiten und Fehlerarten

Reference Broadcast Sync.

RoundTrip Time Sync. (NTP)

Driftberechnung in RBS vs. RTT Sync.

Time Protocol for Sensorn Netw.

Intervallschätzung f. Ereignisse

Vergleiche von Zeitintervallen

## Synchronisation in Sensornetzen

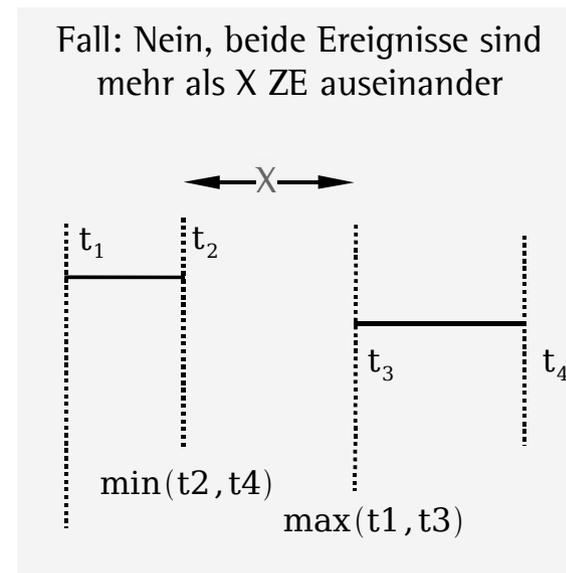
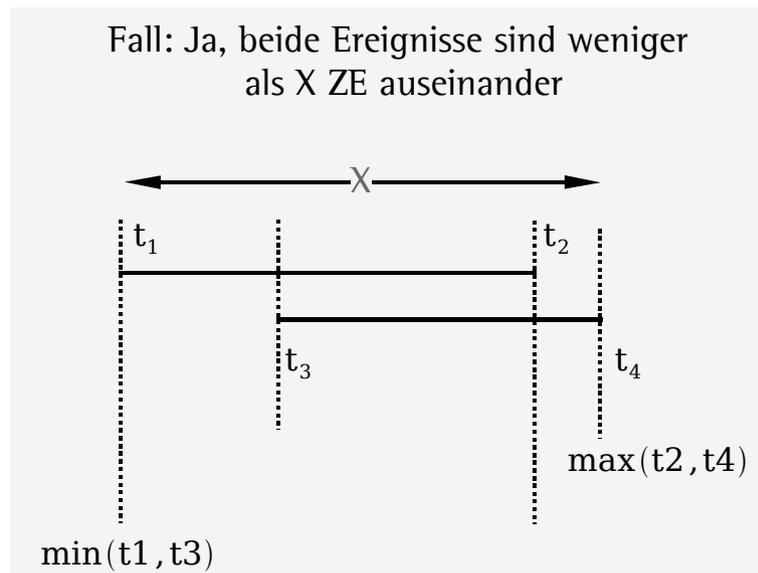
### Schätzung von Zeit-Intervallen in Sensornetzen

Vergleichende Aussagen über Ereignis-Intervalle

Liegen die Ereignisse E1 und E2 weniger als X Zeiteinheiten auseinander?

Ja	wenn $\max(t_4, t_2) - \min(t_3, t_1) < X (1-p)$
Nein	wenn $\max(t_3, t_1) - \min(t_2, t_4) > X (1+p)$
Vielleicht	sonst

Ist bekannt, dass die Genauigkeit der Uhren im Intervall  $[1-p, 1+p]$  liegt, so kann man die Realzeit X je nach Fragestellung an die max. zu erwartende Ungenauigkeit der Uhr anpassen.



Zeiten und Fehlerarten

Reference Broadcast Sync.

RoundTrip Time Sync. (NTP)

Driftberechnung in RBS vs. RTT Sync.

Time Protocol for Sensornetw.

Intervallschätzung f. Ereignisse

Vergleiche von Zeitintervallen