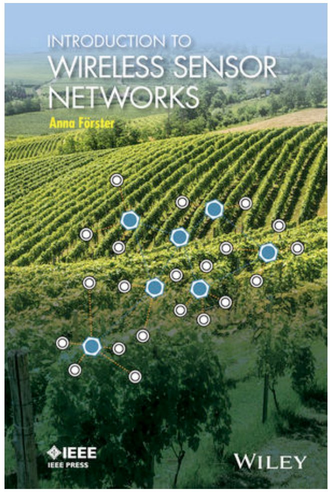


INTRODUCTION TO WIRELESS SENSOR NETWORKS

CHAPTER 7: TIME SYNCHRONIZATION

Anna Förster

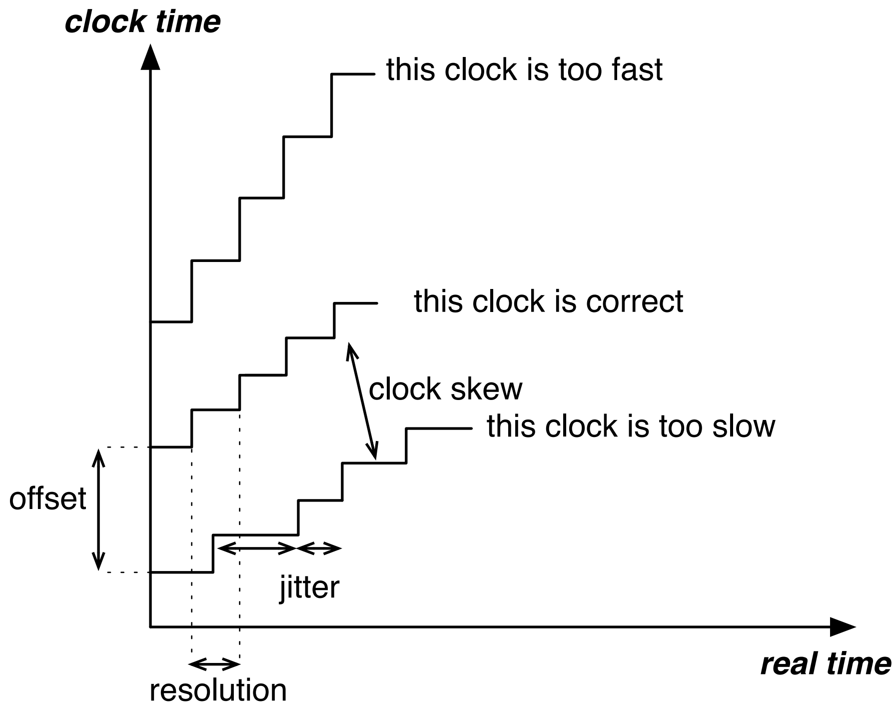
IEEE
IEEE PRESS
WILEY
comnets

OVERVIEW

1. Clocks and Delay Sources
2. Requirements and Challenges
3. Time Synchronization Protocols
    1. Lightweight Tree Synchronization
    2. Reference Broadcast Synchronization
    3. NoTime

Clocks and Delay Sources

- ***Clock resolution or rate*** *is the step at which the clock progresses and can be read*
- ***Clock skew or drift*** *is the difference between the progressing speeds of two clocks*
- ***Clock offset*** *is the difference between the time shown by two different clocks*
- ***Clock jitter*** *is when the individual ticks of a clock are not perfectly the same*



Requirements and Challenges (1)

- **Two clocks drift apart even after synchronization**
- **The maximum drift rate of a clock dictates the synchronization period**

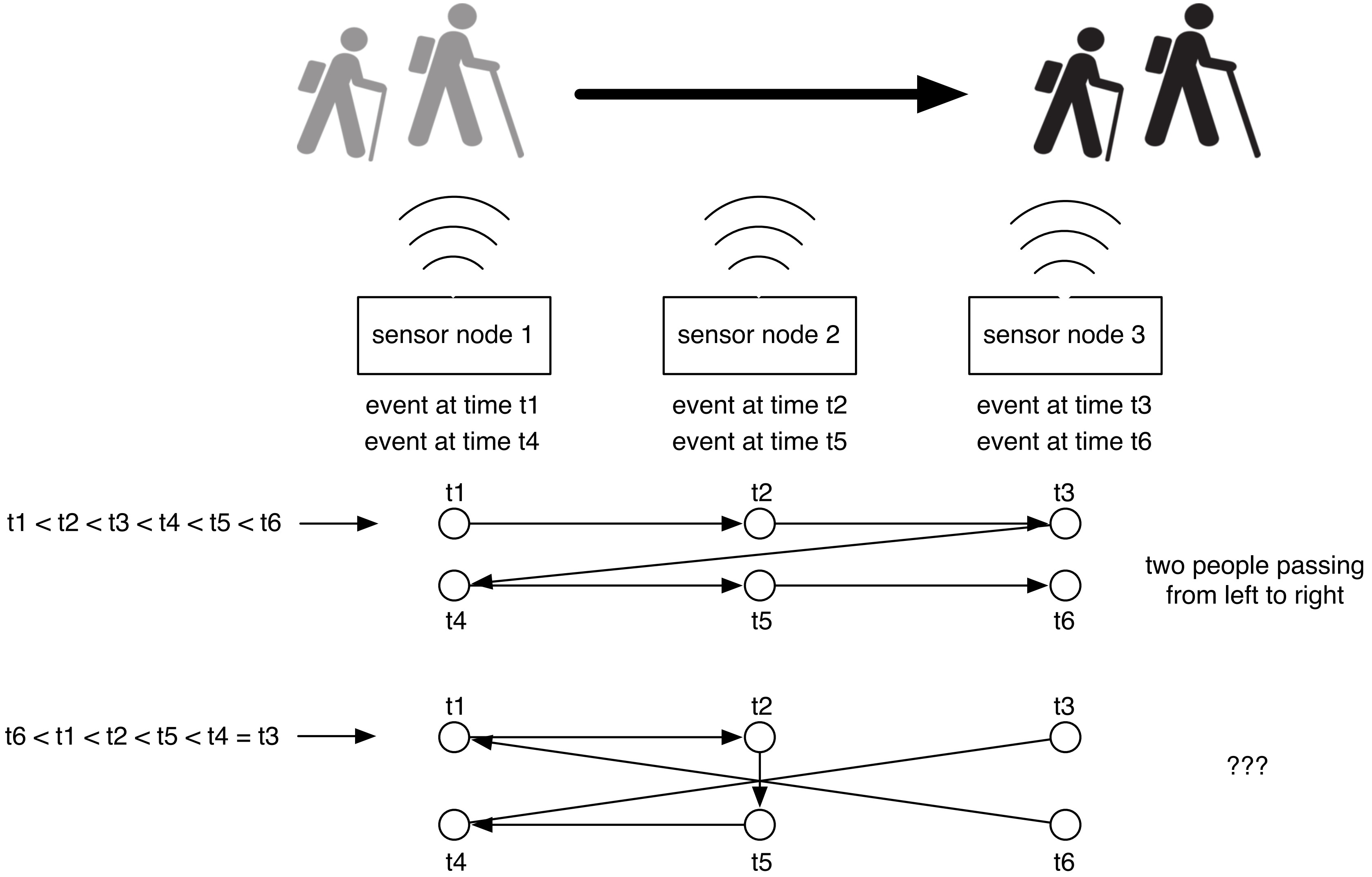
$$\tau_{sync} = \frac{\delta}{2\rho}$$

where ρ is the maximum drift rate in ppm (parts per million), δ is the maximum allowed drift in seconds and τsync is your needed synchronization period.

Requirements and Challenges (2)

- **The time synchronization protocol cannot jump back and forth in time.**
- ***Time synchronization accuracy*** *is the maximum offset between the synchronized clock and a reference clock (for example global time).*
- ***Time synchronization precision*** *is the maximum offset between any two synchronized clocks.*
- **Time synchronization precision is important to understand the sensor data.**

Understanding Sensor Data

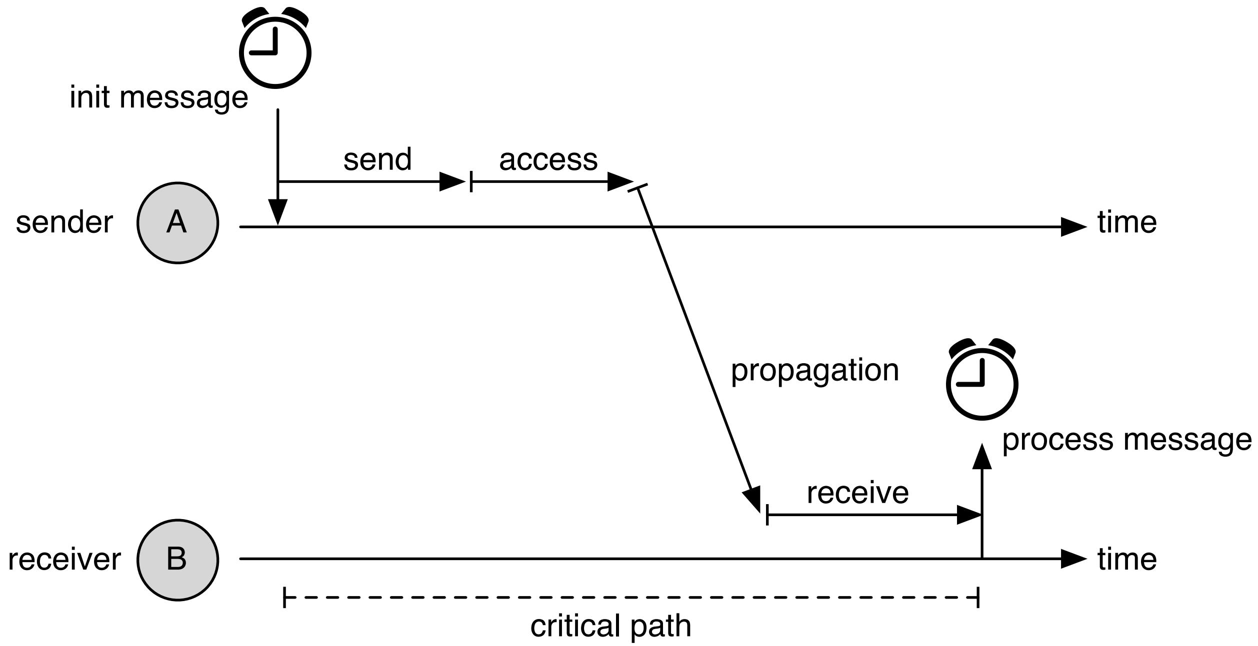


Anna Förster, Introduction to Wireless Sensor Networks, 2016

Time Synchronization Protocols

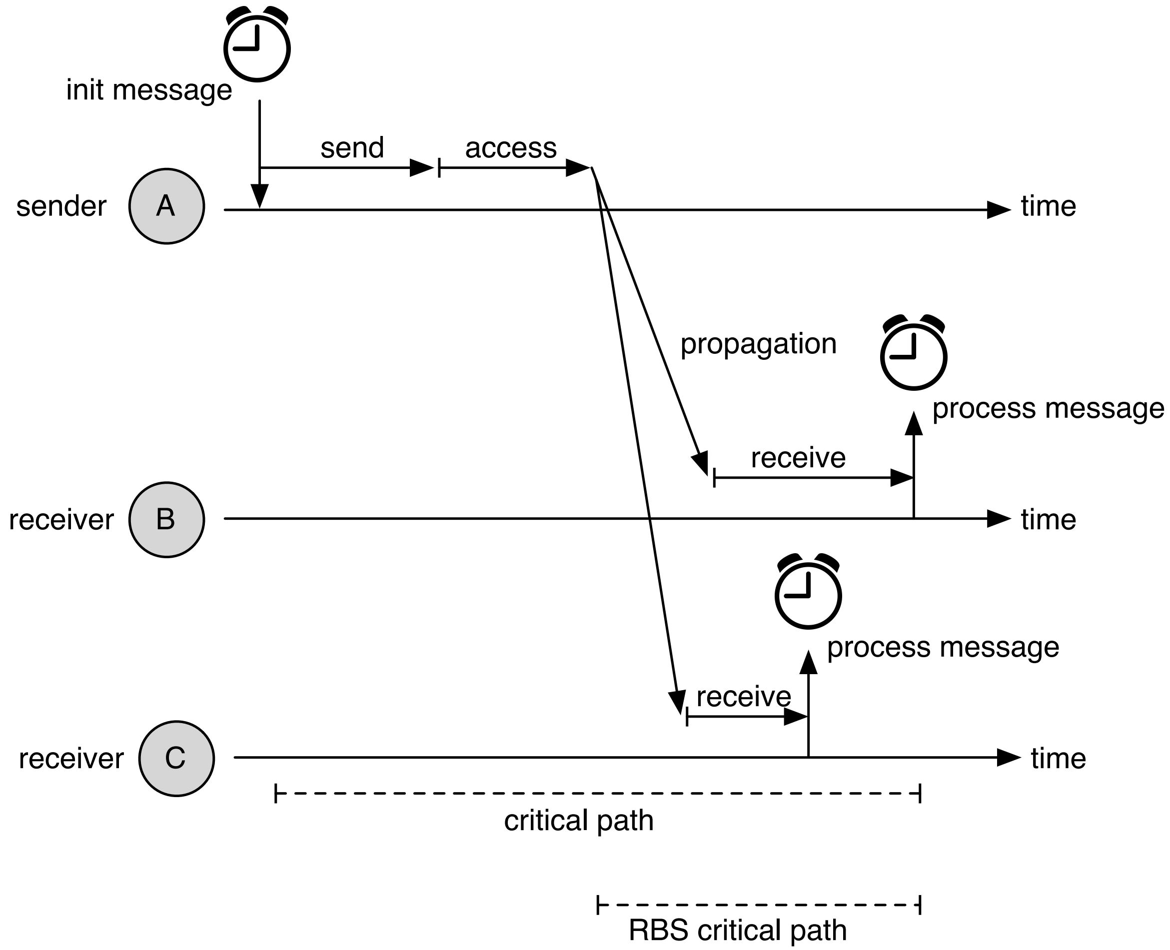
- **Externally** Synchronized: there is an external source of correct time data.
- **Internally** Synchronized: consistent view from all nodes, but not globally correct time.

Sender-Receiver Synchronization



- Long critical path

Receiver-Receiver Synchronization



- Shorter critical path

Lightweight Tree Synchronization (LTS)

- Build a spanning tree from root (sink) to all nodes
- Sink has globally correct time
- Sink sends a message to its children to sync them, then traverse the tree
- Use sender-receiver synchronization

- Precision depends on depth of tree and does not scale

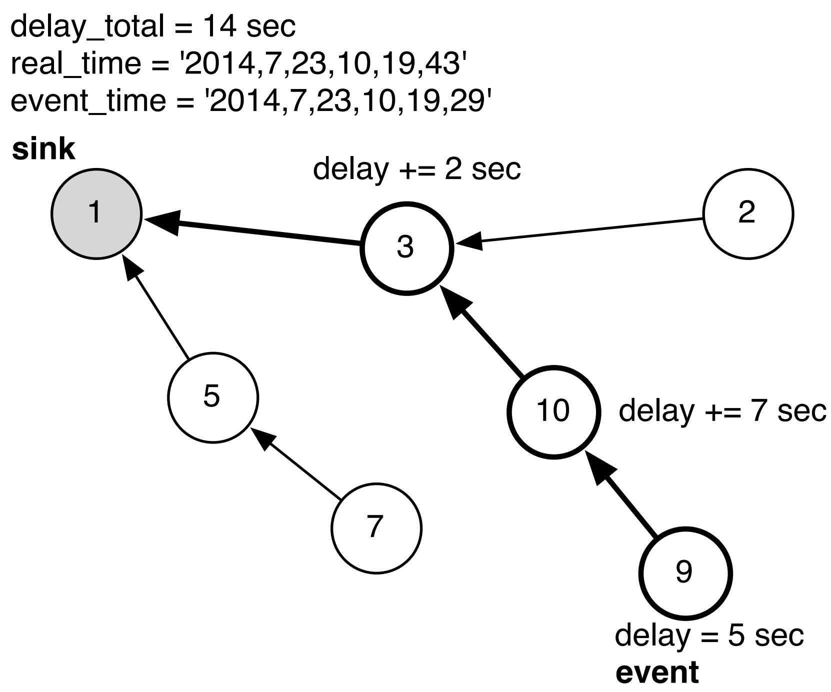
Reference Broadcast Synchronization (RBS)

- Uses receiver-receiver synchronization
- Broadcast correct time to all neighbors

- Total precision increases with number of broadcasts
- Scales well
- Not very efficient in terms of communication overhead

NoTime Protocol

- Do not try to time sync the nodes directly
- Provide enough information so that the sink can reconstruct the real time
- Save delays on the way



- Precision depends on:
  - Propagation Delay
  - Hardware Delay
  - Local time resolution

SUMMARY

- **Clock skew** causes two clocks to progress at different rates, while **clock jitter** causes a clock to tick irregularly.
- Provide efficiency and simplicity:
  - Regular synchronization is needed.
  - Available time cannot jump back and forth in.
- **Lightweight tree synchronization protocol (LTS)** uses the sender-receiver synchronization message exchange and is not scalable.
- **Broadcast reference protocol (RBS)** uses the receiver-receiver synchronization message exchange, has greater precision than LTS and is scalable.
- **NoTime** does not synchronize the nodes, but computes the in-network delay for each data packet, which enables the sink to compute the original event's real time.

IEEE
IEEE PRESS
WILEY
comnets