Clock Synchronization in Wireless Sensor Networks: Local vs. Global

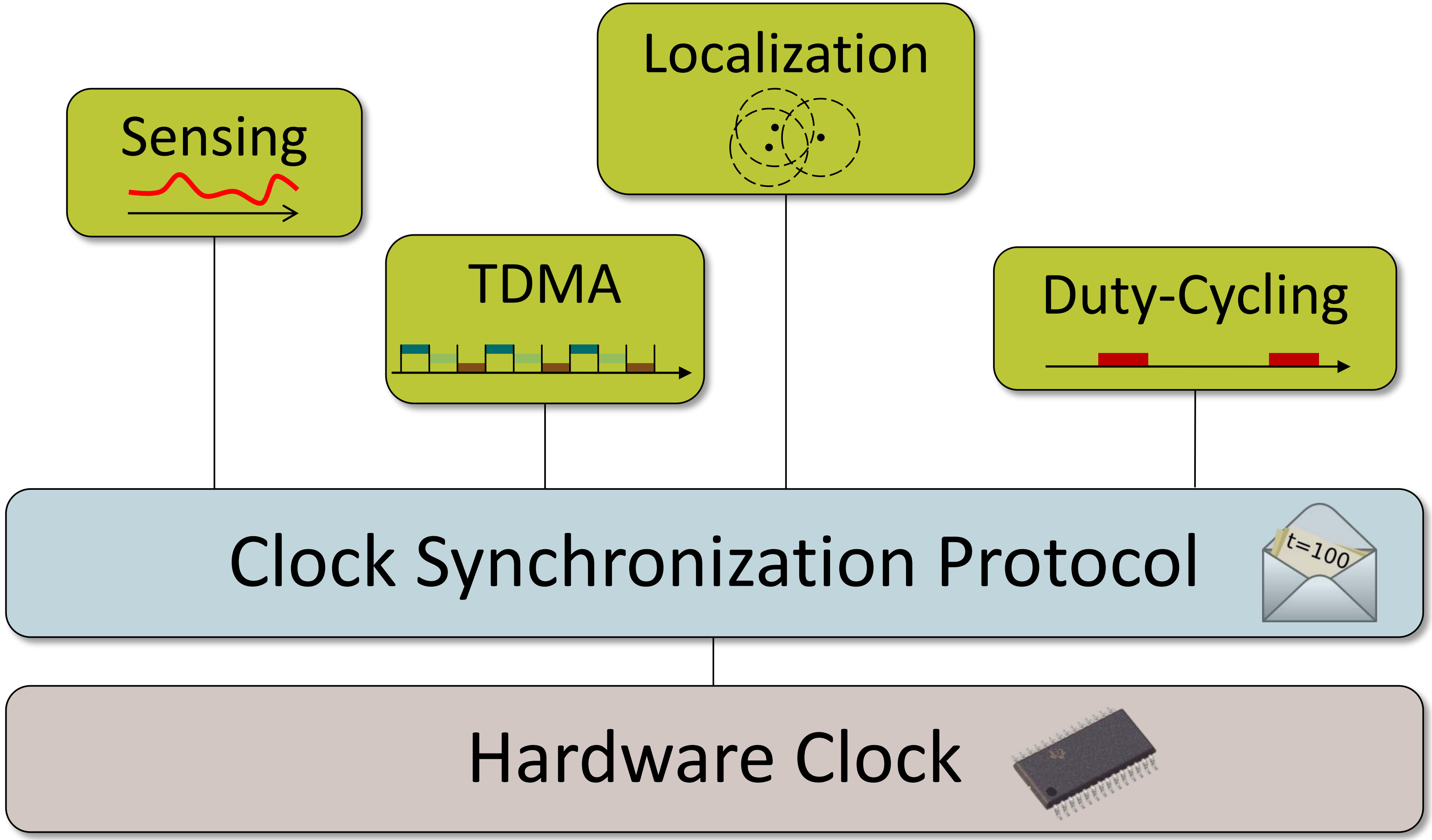
Philipp Sommer
Roger Wattenhofer

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing Group

Time in Sensor Networks

- Synchronized clocks are essential for many applications:



Clock Synchronization in Practice

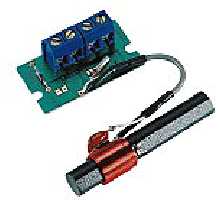
- Many different approaches for clock synchronization

Global Positioning System (GPS)

+ high accuracy
- outdoor use only
- high energy consumption

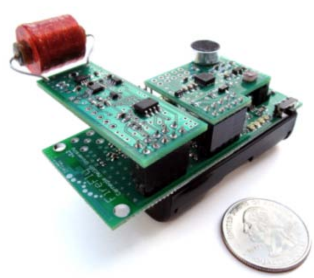
Radio Clock Signal

+ indoor use possible
- low cost, low energy
- limited accuracy
- bulky antenna



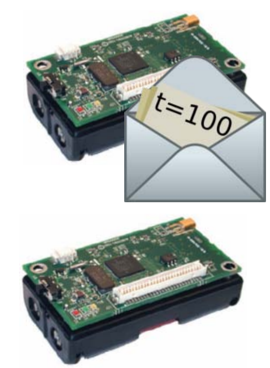
AC-power line radiation

+ low accuracy
+ low energy consumption
- indoor use only
- bulky device



Synchronization messages

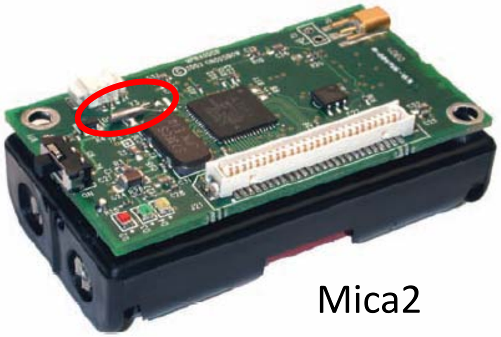
+ high accuracy
+ indoor/outdoor
+ low energy
+ no additonal hardware



Hardware Clocks of Sensor Nodes

- Counter register of the microcontroller

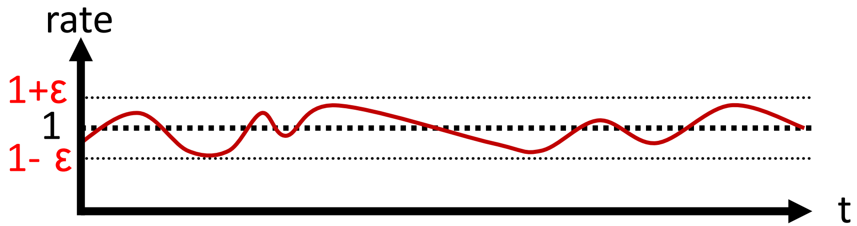
  Sourced by an external crystal (32kHz, 7.37 MHz)



Mica2

- Clock drift

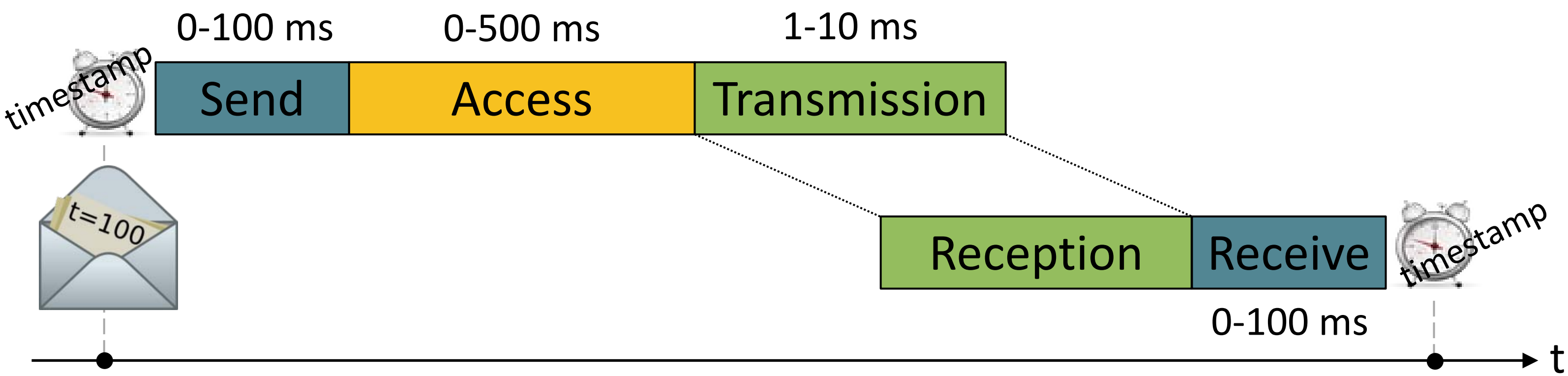
  Random deviation from the nominal rate dependent on ambient temperature, power supply, etc. (30-100 ppm)



Message Delay in Wireless Sensor Networks

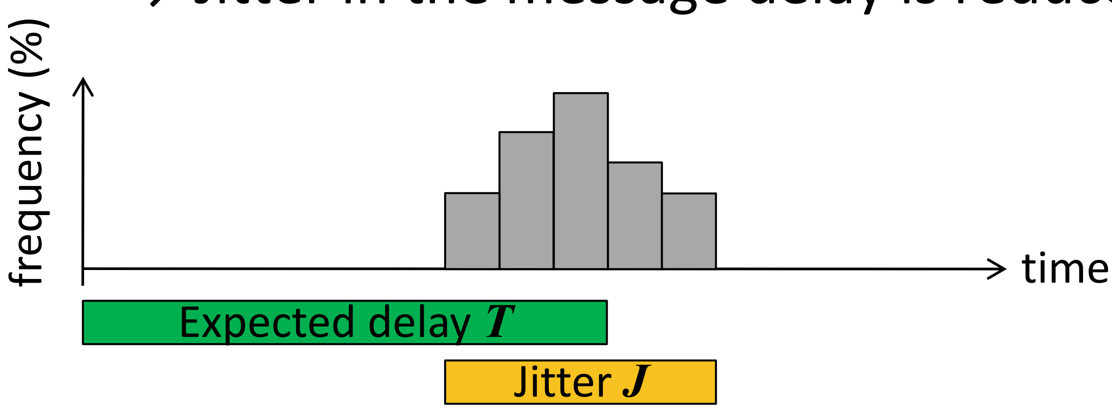
- Problem: Jitter in the message delay

  Various sources of errors (deterministic and non-deterministic)



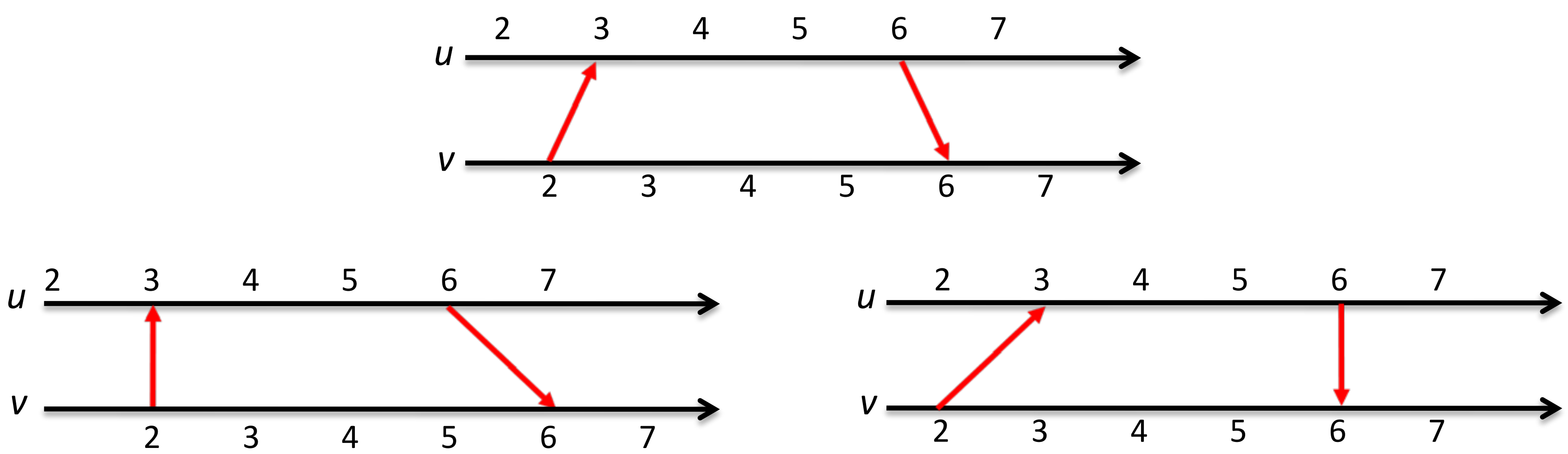
- Solution: Timestamping packets at the MAC layer (Maróti et al.)

  → Jitter in the message delay is reduced to a few clock ticks



Bounds on the Synchronization Accuracy

- Two nodes *u* and *v* cannot be synchronized perfectly

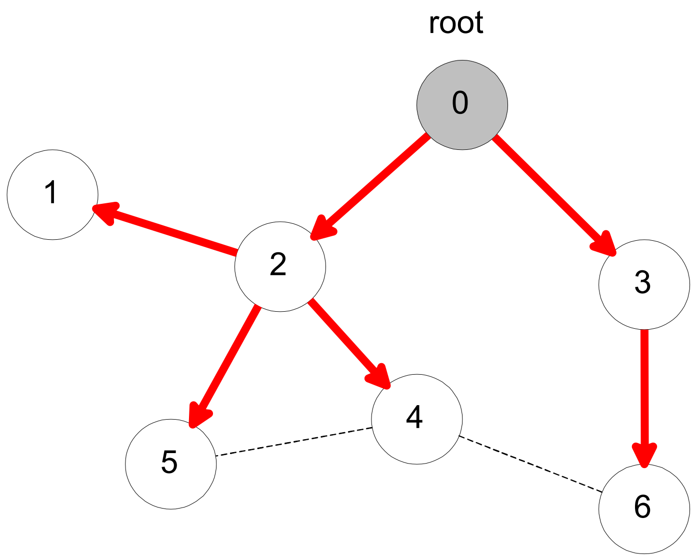


- Messages between two neighboring nodes may be fast in one direction and slow in the other, or vice versa.
- Error increases as the square-root of the distance from the reference node

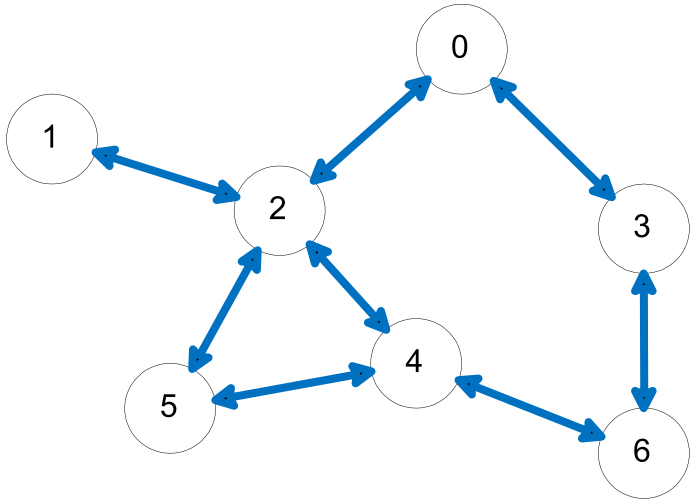
Clock Synchronization: Local vs. Global

- **Global** property: Minimize clock error between **any** two nodes
- **Local** ("gradient") property: Small clock error between two nodes if the **distance** between the nodes is small.

Flooding Time Synchronization Protocol (FTSP)



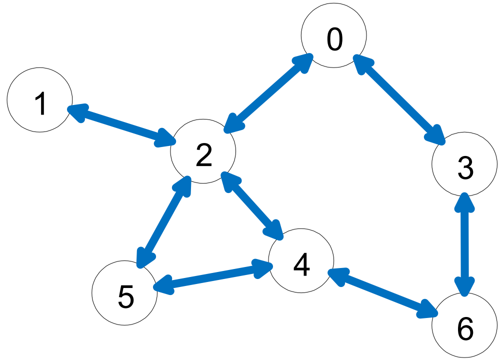
Gradient Time Synchronization Protocol (GTSP)



Gradient Time Synchronization Protocol (GTSP)

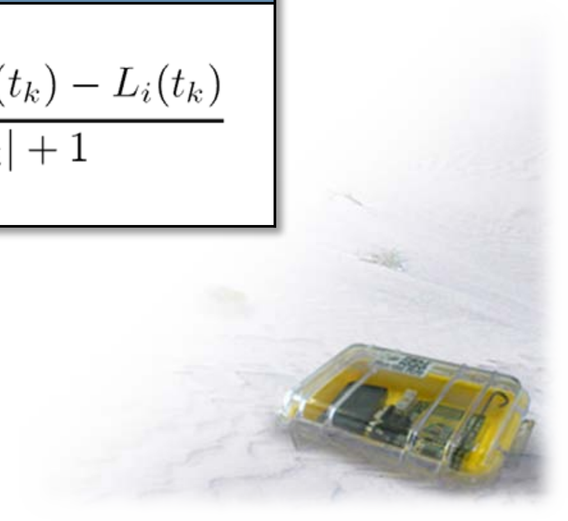
[Sommer et al., IPSN‘09]

- Synchronize clocks with **all** neighboring nodes
  - No reference (root) node necessary
  - No tree or pre-established topology



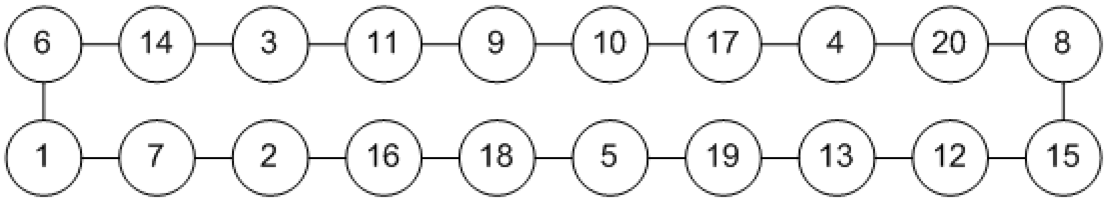
- Averaging clock value/rate of all neighbors (including node itself)

Clock Rate	Clock Offset
$l_i(t_{k+1}) = \frac{\left(\sum_{j \in \mathcal{N}_i} \frac{x_j(t_k)}{h_i(t_k)}\right) + l_i(t_k)}{\lvert\mathcal{N}_i\rvert + 1}$	$\theta_i(t_{k+1}) = \theta_i(t_k) + \frac{\sum_{j \in \mathcal{N}_i} L_j(t_k) - L_i(t_k)}{\lvert\mathcal{N}_i\rvert + 1}$



Experimental Evaluation

- Testbed of 20 Crossbow Mica2 sensor nodes

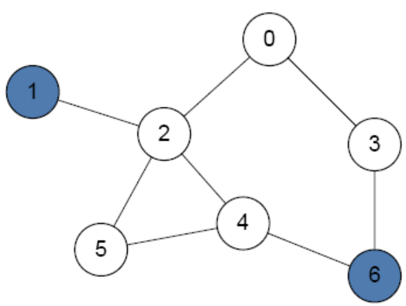


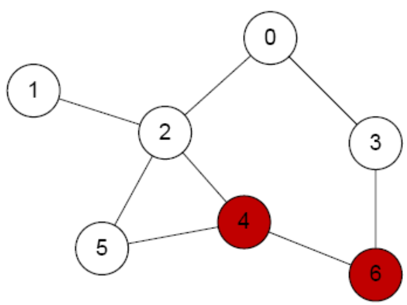
- Global clock synchronization error

  Pair-wise synchronization error between **any** nodes in the network

- Local clock synchronization error

  Pair-wise synchronization error between **neighboring** nodes



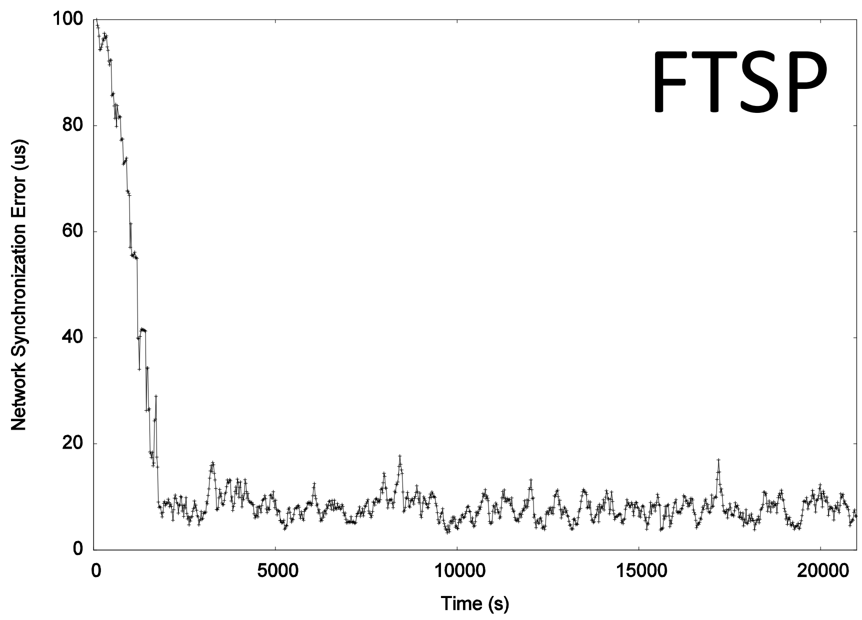


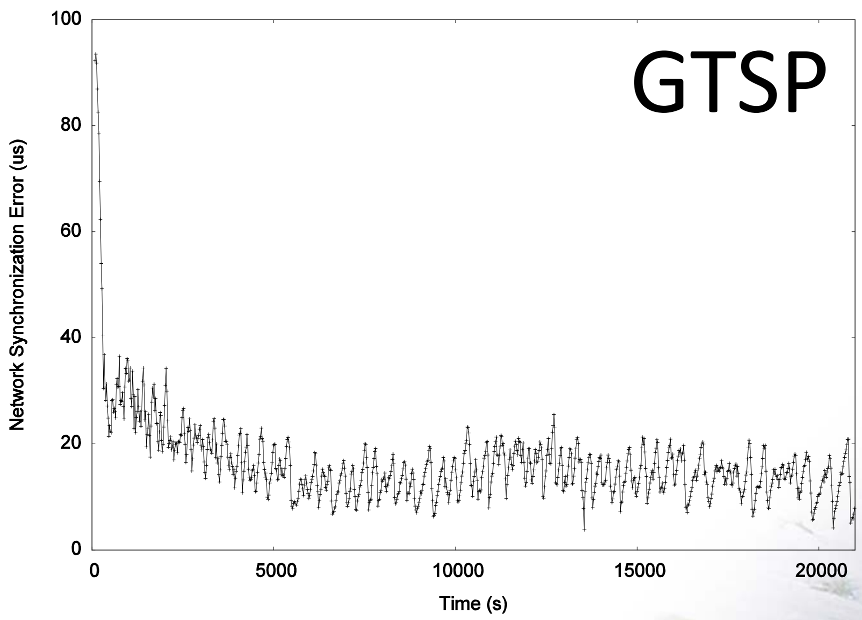
Experimental Results

- Global clock synchronization error

  7.7 µs with FTSP, **14.0** µs with GTSP

  FTSP needs more time to synchronize all nodes after startup



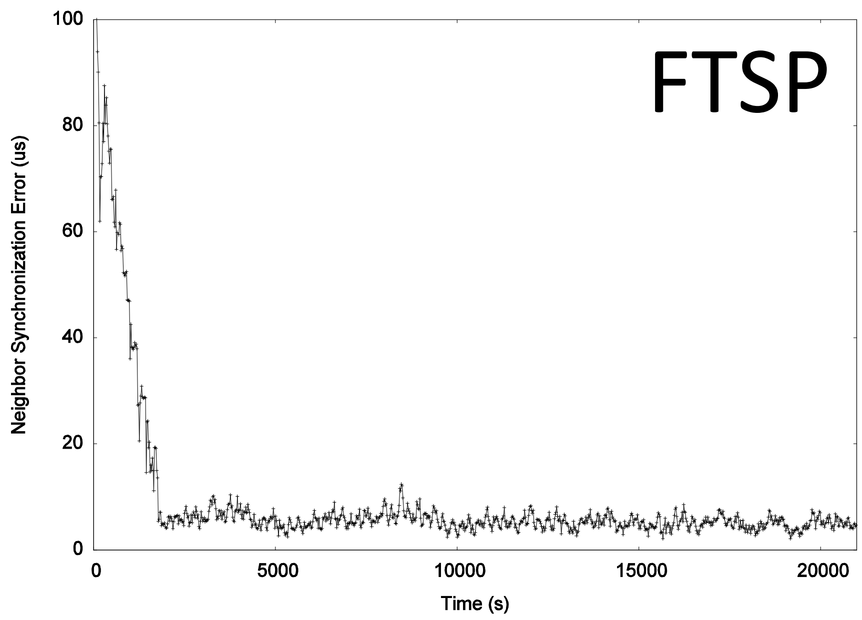


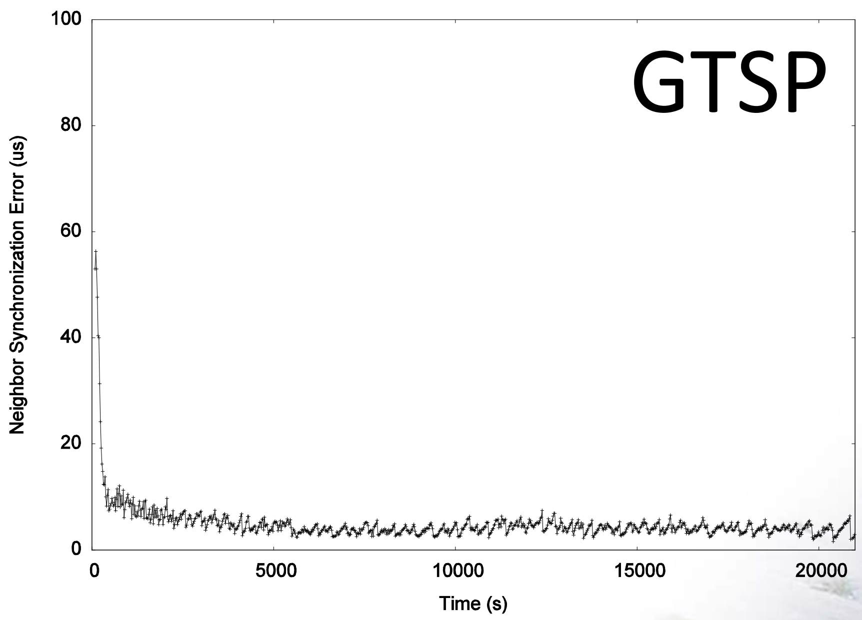
Experimental Results (2)

- Local clock synchronization error

  5.3 μs with FTSP, **4.0** μs with GTSP

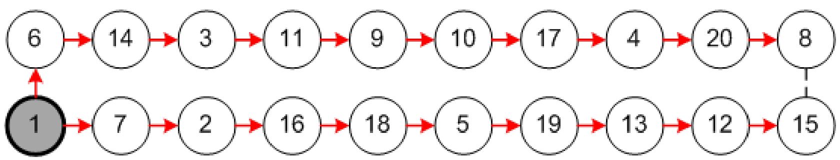
  GTSP takes slightly more time to stabilize

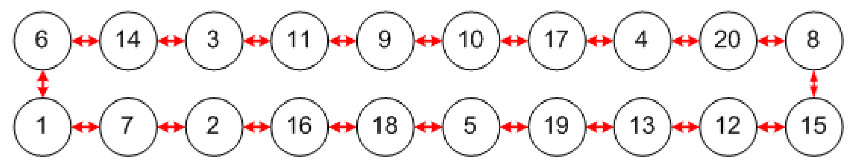


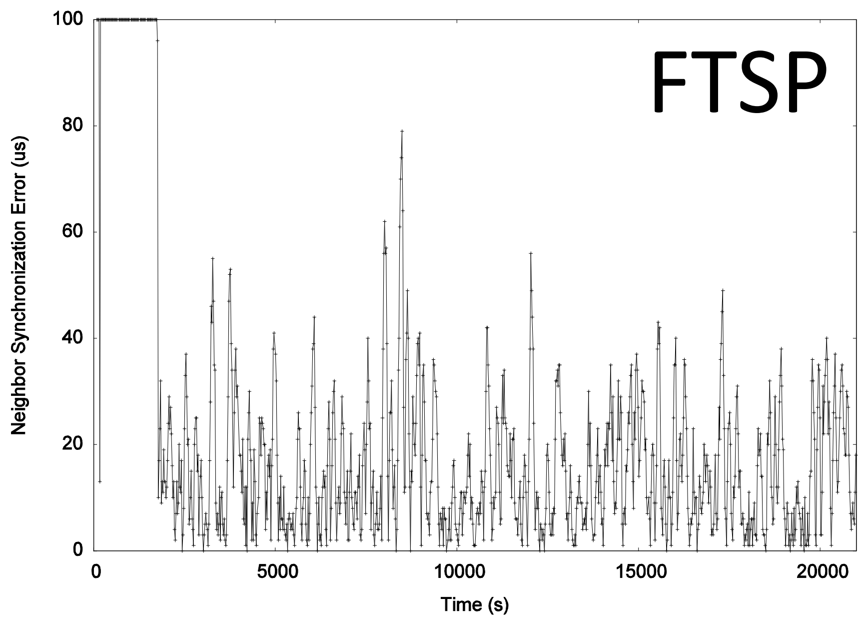


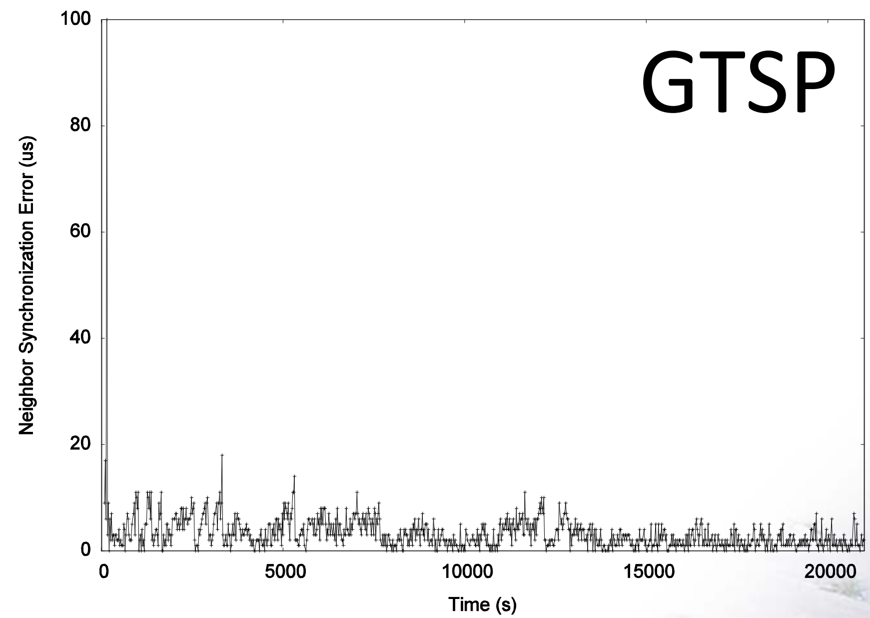
Neighbor Synchronization Error: FTSP vs. GTSP

- FTSP has a large clock error for neighbors with large stretch in the tree (Node 8 and Node 15)



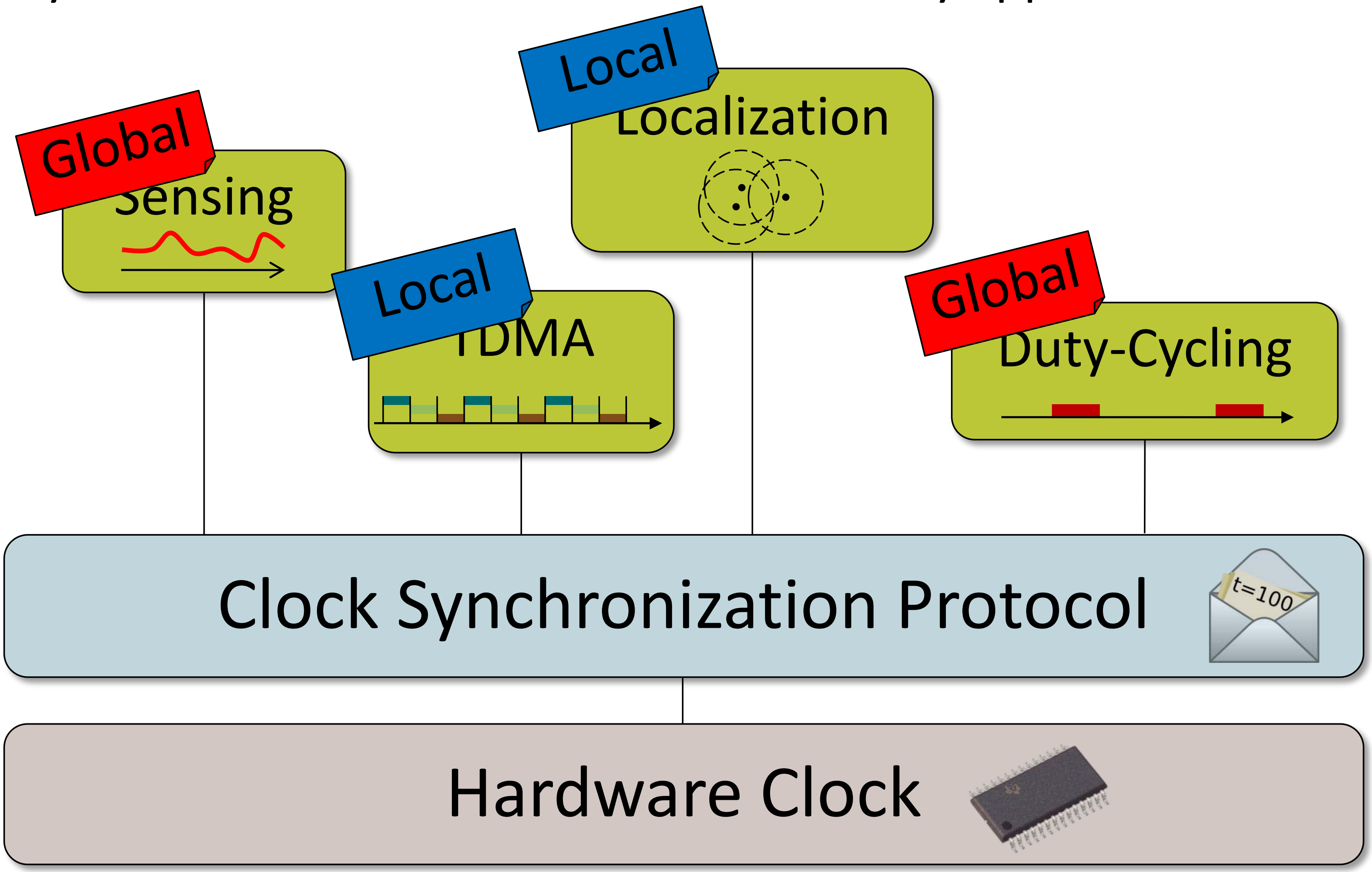




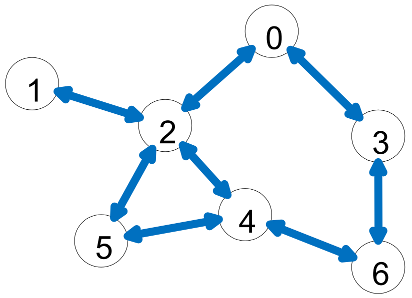


Time in Sensor Networks (Revisited)

- Synchronized clocks are essential for many applications:



Conclusion and Future Work



- Gradient Time Synchronization Protocol (GTSP)
  - Distributed time synchronization algorithm (no leader)
  - Improves the synchronization error between neighboring nodes while still providing precise network-wide synchronization

- Is there a „perfect“ clock synchronization protocol?
  - Goal: Minimizing local and global error at the same time