Wireless Sensor Networks

12th Lecture

05.12.2006

CoNe Freiburg

University of Freiburg
Computer Networks and Telematics
Prof. Christian Schindelhauer

Christian Schindelhauer

schindel@informatik.uni-freiburg.de

University of Freiburg
Institute of Computer Science
Computer Networks and Telematics
Prof. Christian Schindelhauer

Overview

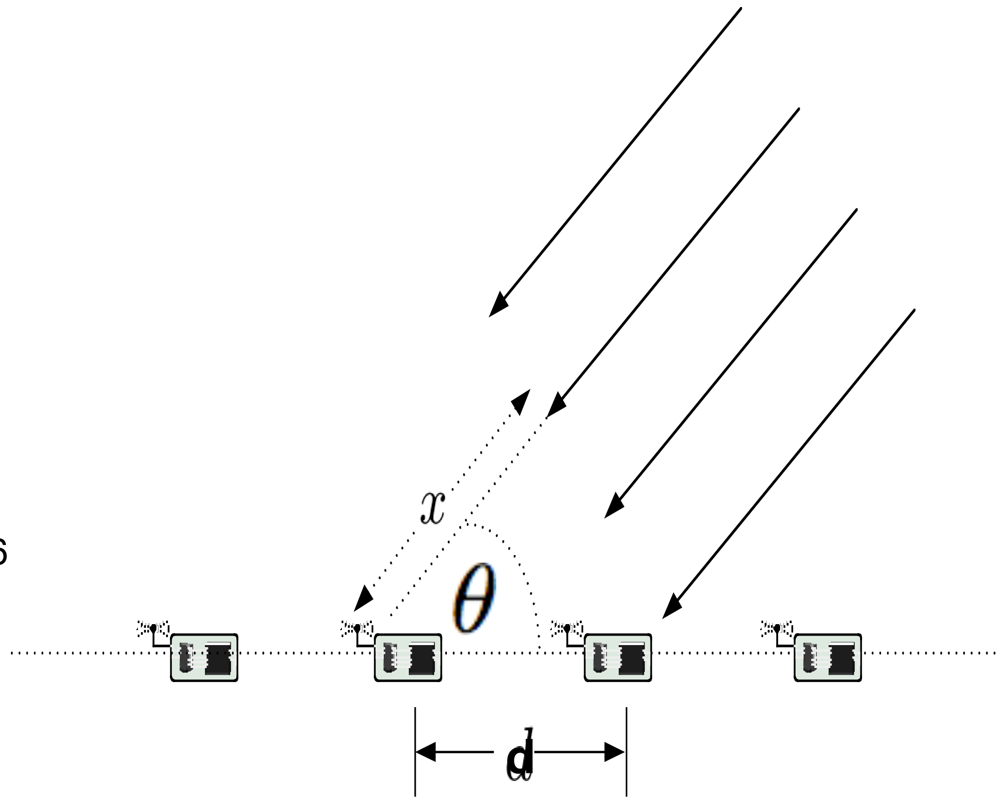
- **The time synchronization problem**
- **Protocols based on sender/receiver synchronization**
- **Protocols based on receiver/receiver synchronization**
- **Summary**

Example

- **Goal: estimate angle of arrival of a very distant sound event using an array of acoustic sensors**
- **From the figure, θ can be estimated when x and d are known:**

$$x = d \sin \theta$$

- **d is known a priori, x must be estimated from differences in time of arrival**
  - $x = C \Delta_t$ where C is the speed of sound
  - For d=1 m and $\Delta_t=0.001$ we get $\theta = 0.336$ radians = 19.3 degree
  - When Δ_t is estimated with 500 μs error, the θ estimates can vary between 0.166 and 0.518 radians (9.5 ... 29 degree)
- **Morale: a seemingly small error in time synch can lead to significantly different angle estimates**

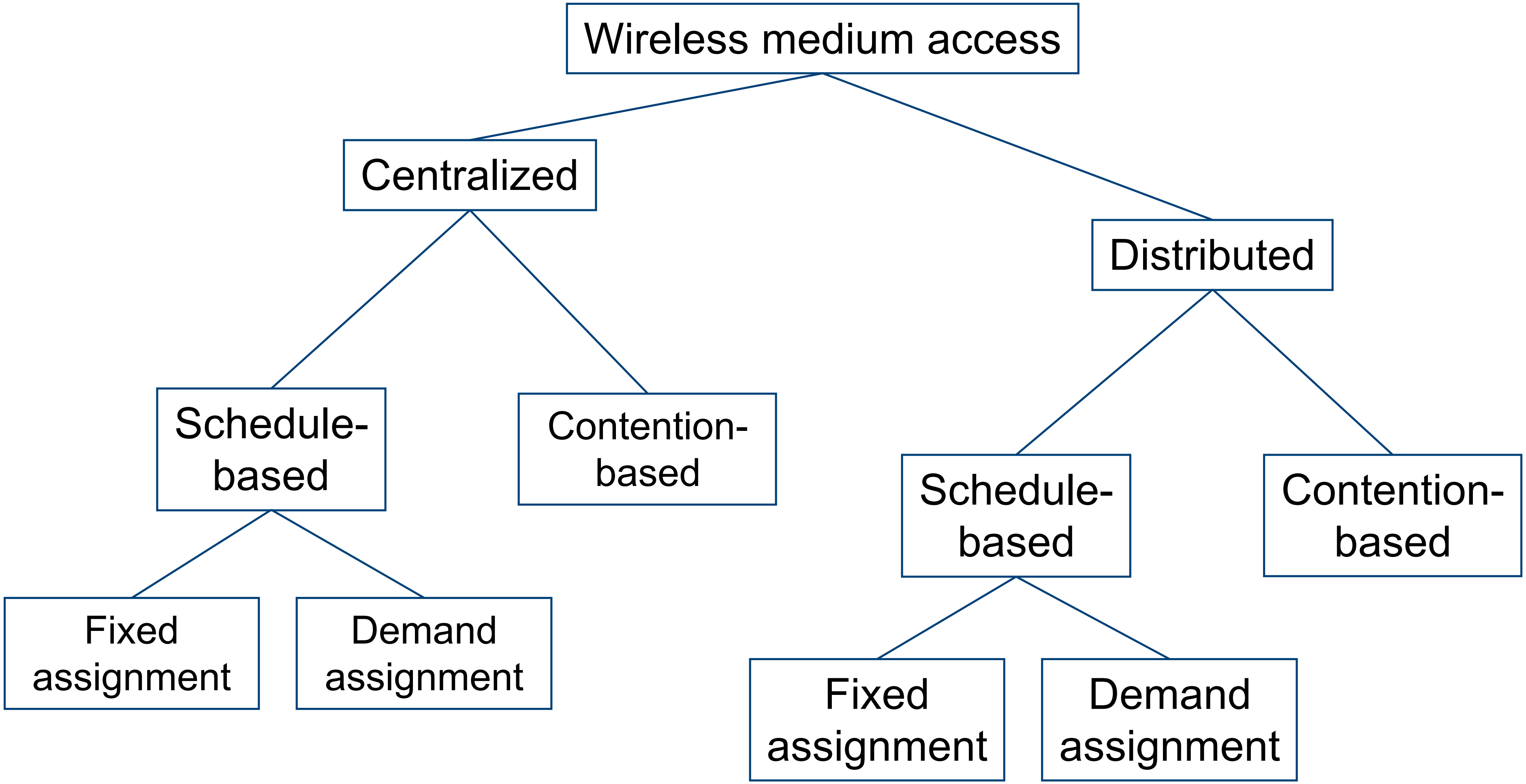


The role of time in WSNs

- **Time synchronization algorithms can be used to better synchronize clocks of sensor nodes**
- **Time synchronization is needed for WSN applications and protocols:**
  - Applications:
    - Arrival of Angle estimation
    - beamforming
  - Protocols:
    - TDMA
    - protocols with coordinated wakeup, ...
  - Distributed debugging
    - timestamping of distributed events is needed to figure out their correct order of appearance

What MAC Relies on Synchronized Clocks?

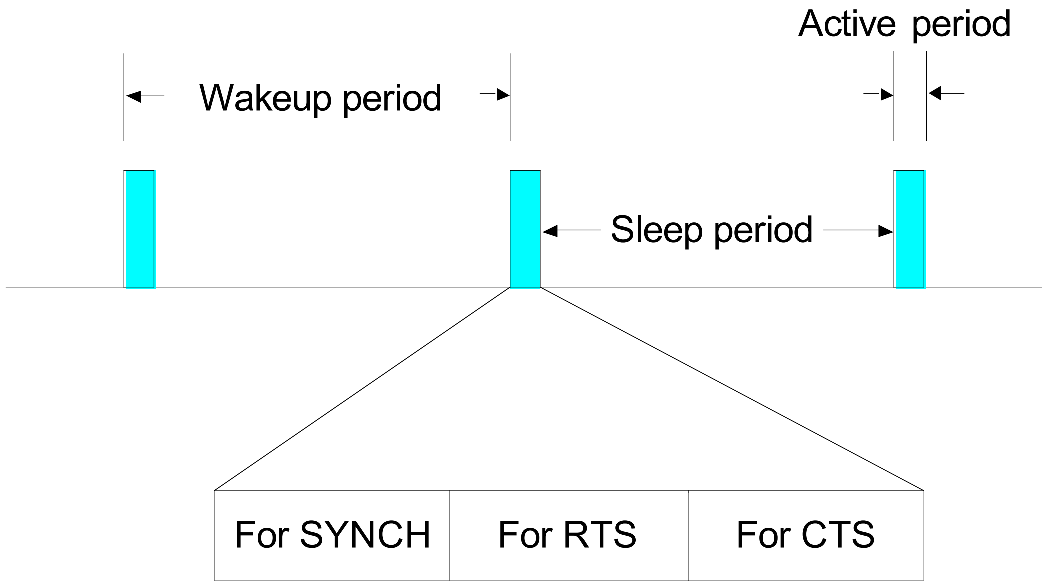
University of Freiburg
Institute of Computer Science
Computer Networks and Telematics
Prof. Christian Schindelhauer



Repetition: Sensor-MAC (S-MAC)

University of Freiburg
Institute of Computer Science
Computer Networks and Telematics
Prof. Christian Schindelhauer

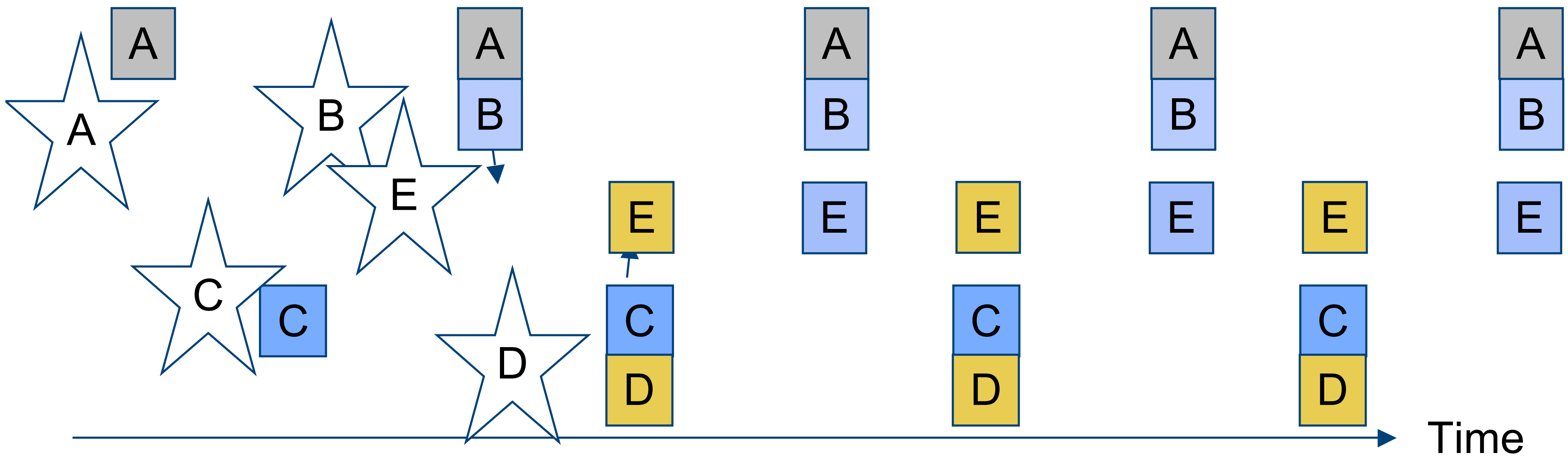
- **MACA's idle listening is particularly unsuitable if average data rate is low**
  - –Most of the time, nothing happens
- **Idea: Switch nodes off, ensure that neighboring nodes turn on simultaneously to allow packet exchange (rendez-vous)**
  - –Only in these ***active periods***, packet exchanges happen
  - –Need to also exchange wakeup schedule between neighbors
  - –When awake, essentially perform RTS/CTS
- **Use SYNCH, RTS, CTS phases**



Repetition: S-MAC synchronized islands

University of Freiburg
Institute of Computer Science
Computer Networks and Telematics
Prof. Christian Schindelhauer

- **Nodes try to pick up schedule synchronization from neighboring nodes**
- **If no neighbor found, nodes pick some schedule to start with**
- **If additional nodes join, some node might learn about two different schedules from different nodes**
  - "Synchronized islands"
- **To bridge this gap, it has to follow both schemes**



Low-Energy Adaptive Clustering Hierarchy (LEACH)

- **Given: dense network of nodes, reporting to a central sink, each node can reach sink directly**
- **Idea: Group nodes into "*clusters*", controlled by** *clusterhead*
  - Setup phase; details: later
  - About 5% of nodes become clusterhead (depends on scenario)
  - Role of clusterhead is rotated to share the burden
  - Clusterheads advertise themselves, ordinary nodes join CH with strongest signal
  - Clusterheads organize
    - CDMA code for all member transmissions
    - TDMA schedule to be used within a cluster
- **In steady state operation**
  - CHs collect & aggregate data from all cluster members
  - Report aggregated data to sink using CDMA

SMACS
Self-Organizing Medium Access Control for Sensor Networks

- **Given: many radio channels, super-frames of known length (not necessarily in phase, but still time synchronization required!)**
- **Goal: set up directional *links* between neighboring nodes**
  - Link: radio channel + time slot at both sender and receiver
  - Free of collisions at receiver
  - Channel picked randomly, slot is searched greedily until a collision-free slot is found
- **Receivers sleep and only wake up in their assigned time slots, once per superframe**
- **In effect: a local construction of a schedule**

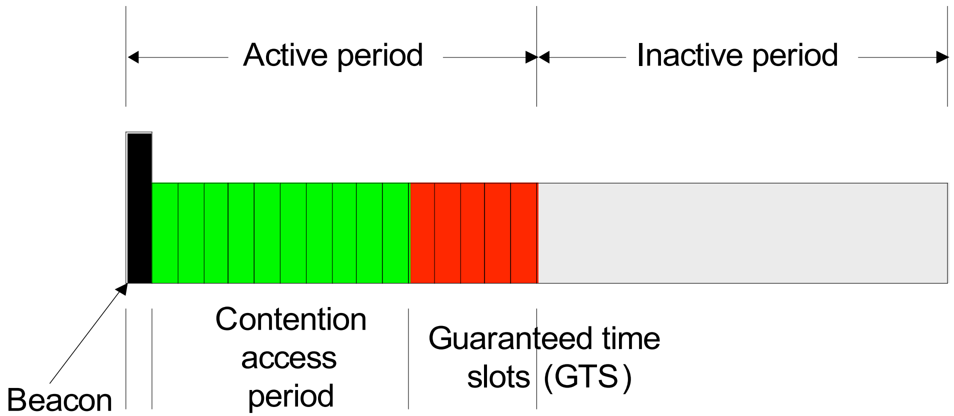
TRAMA
Traffic Adaptive Medium Access Protocol

- **Nodes are synchronized**
- **Time divided into cycles, divided into**
  - Random access periods
  - Scheduled access periods
- **Nodes exchange neighborhood information**
  - Learning about their two-hop neighborhood
  - Using ***neighborhood exchange protocol***: In random access period, send small, incremental neighborhood update information in randomly selected time slots
- **Nodes exchange schedules**
  - Using ***schedule exchange protocol***
  - Similar to neighborhood exchange
- **Adaptive Election Protocol**
  - Elect transmitter, receiver and stand-by nodes for each transmission slot
  - Remove nodes without traffic from election

University of Freiburg
Institute of Computer Science
Computer Networks and Telematics
Prof. Christian Schindelhauer

IEEE 802.15.4 MAC needs Synchronized Clocks

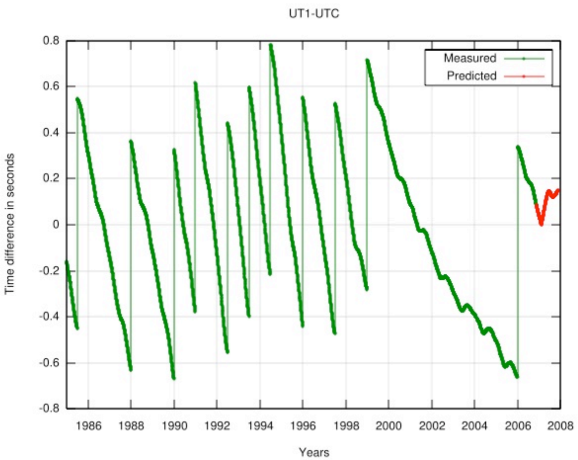
- **Star networks:** *devices* **are associated with** *coordinators*
  - Forming a PAN, identified by a PAN identifier
- **MAC protocol**
  - Single channel at any one time
  - Combines contention-based and schedule-based schemes
- **Beacon-mode superframe structure**
  - GTS assigned to devices upon request



University of Freiburg
Institute of Computer Science
Computer Networks and Telematics
Prof. Christian Schindelhauer

The role of time in WSNs

- **WSN have a direct coupling to the physical world,**
  - notion of time should be related to ***physical time***:
- **physical time = wall clock time, real-time**
  - one second of a WSN clock should be close to one second of real time
- **Commonly agreed time scale for real time is UTC**
  - Coordinated Universal Time
  - generated from atomic clocks
  - modified by insertion of leap seconds to keep in synch with astronomical timescales (one rotation of earth)
- **Universal Time (UT)**
  - timescale based on the rotation of earth
- **Other concept: logical time (Lamport**
  - relative ordering of events counts but not their relation to real time



Clocks in WSN nodes

- **Often, a *hardware clock* is present:**
  - Oscillator generates pulses at a fixed nominal frequency
  - A counter register is incremented after a fixed number of pulses
    - Only register content is available to software
    - Register change rate gives achievable time resolution
  - Node i's register value at real time t is $H_i(t)$
    - Convention: small letters (like t, t') denote real physical times, capital letters denote timestamps or anything else visible to nodes

- **A (node-local) software clock is usually derived as follows:**

$$L_i(t) = \theta_i H_i(t) + \phi_i$$

    - (not considering overruns of the counter-register)
  - θ_i is the (drift) rate, ϕ_i the phase shift
  - Time synchronization algorithms modify θ_i and ϕ_i, but not the counter register

Synchronization accuracy / agreement

- **External synchronization:**
  - synchronization with external real time scale like UTC
  - Nodes i=1, ..., n are accurate at time t within bound δ when

$$|L_i(t) - t| < \delta \text{ for all } i$$

    - Hence, at least one node must have access to the external time scale
- **Internal synchronization**
  - No external timescale, nodes must agree on common time
  - Nodes i=1, ..., n agree on time within bound δ when

$$|L_i(t) - L_j(t)| < \delta \text{ for all } i,j$$

Sources of inaccuracies

- **Nodes are switched on at random times**
  - phases θ_i are random
- **Actual oscillators have random deviations from nominal frequency**
  - (drift, skew)
- **Deviations are specified in ppm (pulses per million)**
  - the ppm value counts the additional pulses or lost pulses over the time of one million pulses at nominal rate
- **The cheaper the oscillators, the larger the average deviation**
  - For sensor nodes
    - values between 1 ppm (one second every 11 days) 100 ppm (one second every 2.8 hours) are assumed
  - Berkeley motes have an average drift of 40 ppm

Sources of inaccuracies

- **Oscillator frequency depends**
  - on time
    - oscillator aging and
  - environment
    - temperature
    - pressure
    - supply voltage, ...
- **Time-dependent drift rates are not sufficient**
  - frequent re-synchronization necessary
  - However, stability over tens of minutes is often a reasonable assumption

General properties of time synchronization algorithms

University of Freiburg
Institute of Computer Science
Computer Networks and Telematics
Prof. Christian Schindelhauer

- **Physical time versus logical time**
- **External versus internal synchronization**
- **Global versus local algorithms**
  - Keep all nodes of a WSN synchronized or only a local neighborhood?
- **Absolute versus relative time**
- **Hardware versus software-based mechanisms**
  - A GPS, Galileo, GLONASS receiver would be a hardware solution
  - German Broadcasts: A time signal from DCF77
    - Mainflingen, an atomic clock near Frankfurt at about 50.01′N 9.00′E can be received on 77.5 kHz to a range of about 2000 km.
  - Loran-C sends signals for synchronization
  - but often too
    - heavyweight
    - costly
    - energy-consuming in WSN nodes
    - line-of-sight to at least four satellites is required

General properties of time synchronization algorithms

- **A-priori vs. a-posteriori synchronization**
  - Is time synchronization achieved before or after an interesting event?
    - ➔ Post-facto synchronization
- **Deterministic vs. stochastic precision bounds**
- **Local clock update discipline**
  - Should backward jumps of local clocks be avoided?
    - Version control)
  - Avoid sudden jumps?

Performance metrics

- **Precision:**
  - Deterministic algorithms:
    - maximum synchronization error for deterministic algorithms,
  - Stochastic algorithms
    - error mean
    - standard deviation
    - quantiles for stochastic ones
- **Energy costs**
  - # of exchanged packets
  - computational costs
- **Memory requirements**
- **Fault tolerance: what happens when nodes die?**

Fundamental Building Blocks

- **Resynchronization event detection block:**
  - when to trigger a time synchronization round?
    - Periodically or after external event
- **Remote clock estimation block**
  - figuring out the other nodes clocks with the help of exchanging packets
- **Clock correction block**
  - compute adjustments for own local clock based on estimated clocks of other nodes
- **Synchronization mesh setup block**
  - figure out which node synchronizes with which other nodes

Thank you

(and thanks go also to Andreas Willig for providing slides)

CoNe Freiburg

University of Freiburg
Computer Networks and Telematics
Prof. Christian Schindelhauer

Wireless Sensor Networks
Christian Schindelhauer
schindel@informatik.uni-freiburg.de

12th Lecture
05.12.2006