Wireless Sensor Networks

15th Lecture

13.12.2006

Christian Schindelhauer

schindel@informatik.uni-freiburg.de

CoNe Freiburg

University of Freiburg
Computer Networks and Telematics
Prof. Christian Schindelhauer

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

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

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

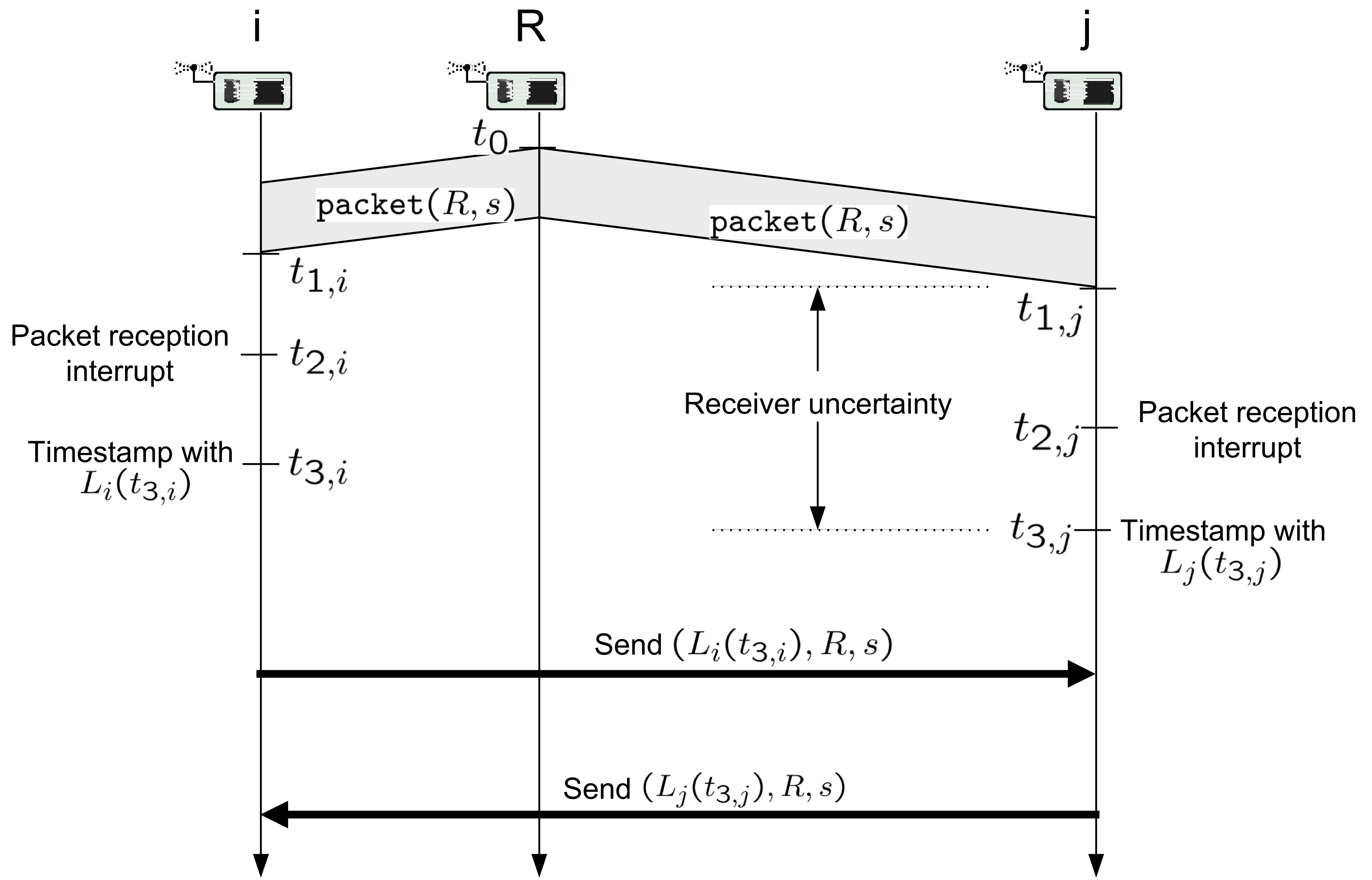
Overview

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

Protocols based on receiver/receiver synchronization

- **Receivers of packets synchronize among each other**
  - not with the transmitter of the packet
- **RBS: Reference Broadcast Synchronization**
  - Elson, Girod, Estrin, [OSDI 2002]
  - Synchronize receivers within a single broadcast domain
  - A scheme for relating timestamps between nodes in different domains
- **RBS**
  - does not modify the local clocks of nodes
  - but computes a table of conversion parameters for each peer in a broadcast domain
  - allows for post-facto synchronization

RBS – Synchronization in a Broadcast Domain



RBS – Synchronization in a Broadcast Domain

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

- **The goal is to synchronize i's and j's clocks to each other**
- **Timeline:**
  - Reference node R broadcasts at time t_0 some synchronization packet carrying its identification R and a sequence number s
  - Receiver i receives the last bit at time $t_{1,i}$, gets the packet interrupt at time $t_{2,i}$ and timestamps it at time $t_{3,i}$
  - Receiver j is doing the same
  - At some later time node i transmits its observation ($L_i(t_{3,i})$, R, s) to node j
  - At some later time node j transmits its observation ($L_j(t_{3,j})$, R, s) to node i
  - The whole procedure is repeated periodically, the reference node transmits its synchronization packets with increasing sequence numbers
    - R could also use ordinary data packets as long as they have sequence numbers ...
- **Under the assumption $t_{3,i} = t_{3,j}$ node j can figure out the offset $O_{i,j} = L_j(t_{3,j}) - L_i(t_{3,i})$ after receiving node i's final packet – of course, node i can do the same**

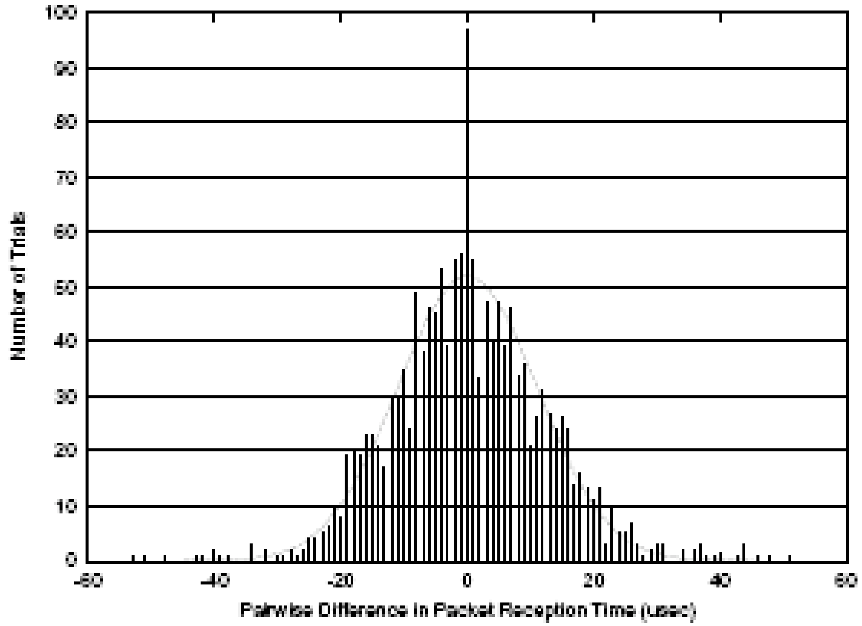
RBS – Synchronization in a Broadcast Domain

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

- **The synchronization error in this scheme can have two causes:**
  - There is a difference between $t_{3,i}$ and $t_{3,j}$
  - Drift between $t_{3,i}$ and the time where node i transmits its observations to j
- **But:**
  - In small broadcast domains and when received packets are timestamped as early as possible the difference between $t_{3,i}$ and $t_{3,j}$ is very small
    - As compared to sender-/receiver based schemes the MAC delay and operating system delays experienced by the reference node play no role!!
  - Drift can be neglected when observations are exchanged quickly after reference packets
  - Drift can be estimated jointly with Offset O when a number of periodic observations of $O_{i,j}$ have been collected
    - This amounts to a standard least-squares line regression problem

RBS – Synchronization in a Broadcast Domain

- **Elson et al**
  - measured pairwise differences in timestamping times at a set of receivers
  - when timestamping happens in the interrupt routine (Berkeley motes)
- **This is just the distribution of the differences $t_{3,i}-t_{3,j}$**



RBS – Synchronization in a Broadcast Domain

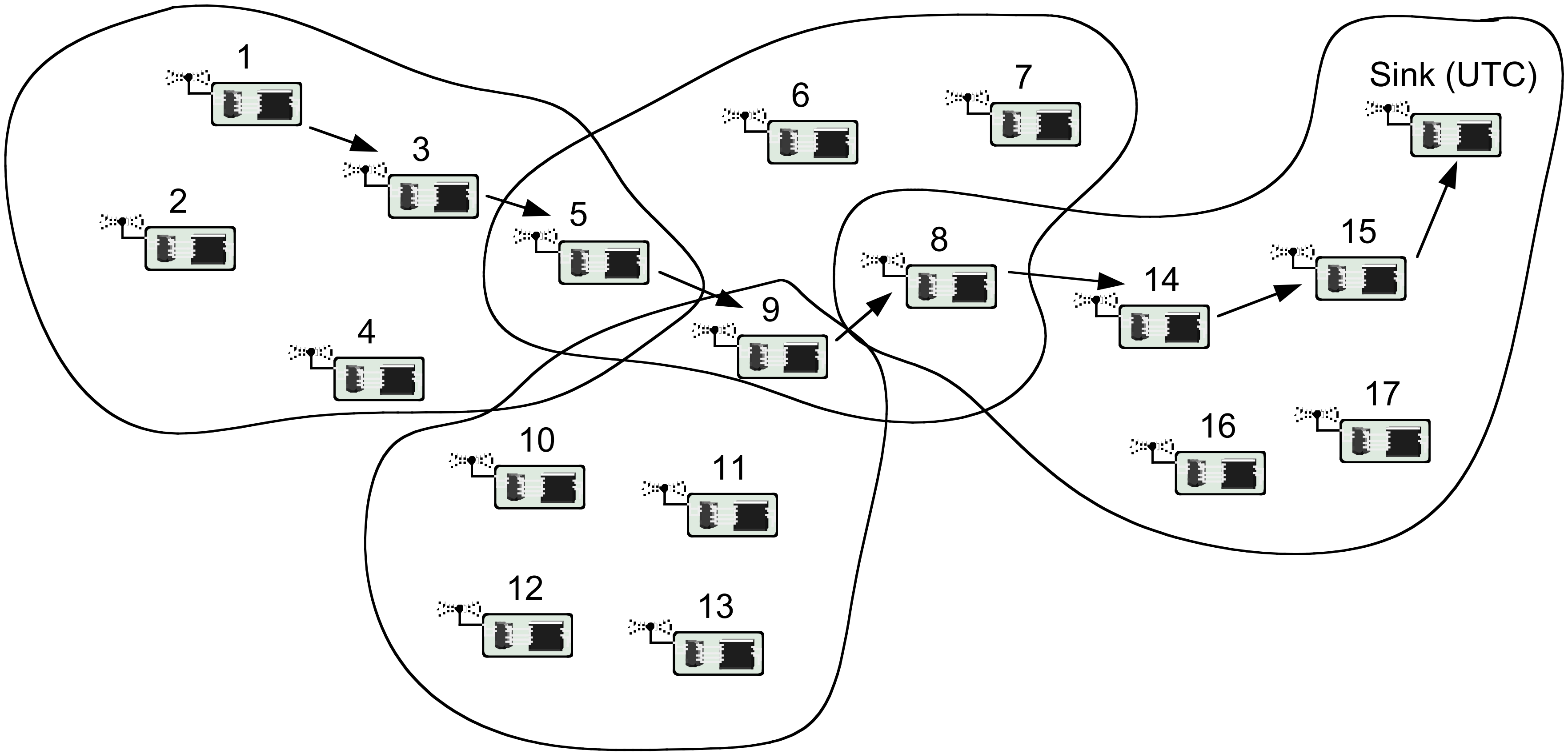
- **Communication costs:**
  - Be n the number of nodes in the broadcast domain
  1. scheme: reference node collects the observations of the nodes, computes the offsets and sends them back

     ➔ 2 n packets
  2. scheme: reference node collects the observations of the nodes, computes the offsets and keeps them, but has responsibility for timestamp conversions and forwarder selection

     ➔ n packets
  3. scheme: each node transmits its observation individually to the other members of the broadcast domain
     - ➔ n (n-1) packets
  4. scheme: each node broadcasts its observation
     - ➔ n packets, but unreliable delivery
- **Collisions:**
  - The reference packets trigger all nodes simultaneously
- **Computational costs**
  - least-squares approximation is not cheap!

RBS – Network Synchronization

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



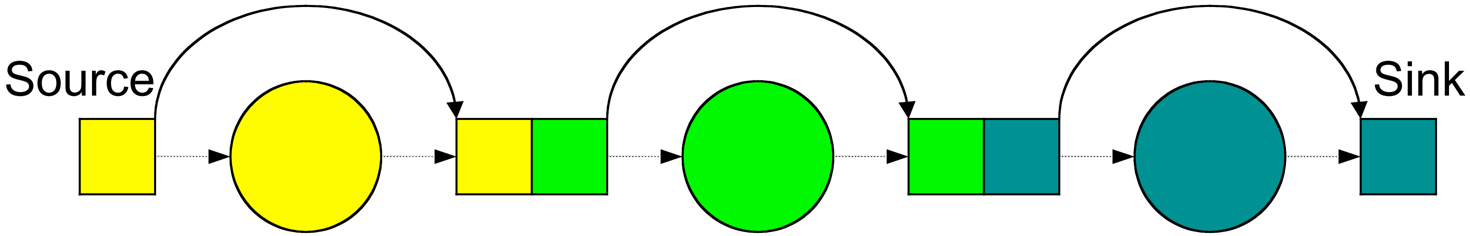
RBS – Network Synchronization

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

- **Suppose that:**
  - node 1 has detected an event at time $L_1(t)$
  - the sink is connected to a GPS receiver and has UTC timescale
  - node 1 wants to inform the sink about the event such that the sink receives a timestamp in UTC timescale
  - Broadcast domains are indicated by "circles"
- **Timestamp conversion approach:**
  - **Idea**: do not synchronize all nodes to UTC time, but convert timestamps as packet is forwarded from node 1 to the sink
    - ➔ avoids global synch
  - Node 1 picks node 3 as forwarder – as they are both in the same broadcast domain, node 1 can convert the timestamp $L_1(t)$ into $L_3(t)$
  - Node 3 picks node 5 in the same way
  - Node 5 is member in two broadcast domains and knows also the conversion parameters for the next forwarder 9
  - And so on ...
  - Result: the sink receives a timestamp in UTC timescale!
  - Nodes 5, 8 and 9 are gateway nodes!

RBS – Network Synchronization

- **Forwarding options:**
  - Let each node pick its forwarder directly and perform conversion, the reference nodes act as mere pulse senders
  - Let each node transmit its packet with timestamp to reference node, which converts timestamp and picks forwarder
    - This way a broadcast domain is not required to be fully connected
  - In either case the clock of the reference nodes is unimportant



- **How to create broadcast domains?**
  - In large domains (large m) more packets have to be exchanged
  - In large domains fewer domain-changes have to be made end-to-end, which in turn reduces synchronization error
  - This is essentially a clustering problem, forwarding paths and gateways have to be identified by routing mechanisms

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

Summary

- **Time synchronization**
  - important for both WSN applications and protocols
  - Using hardware like GPS receivers is typically not an option, so extra protocols are needed
- **Post-facto synchronization**
  - allows time-synchronization on demand
  - otherwise clock drifts would require frequent re-synchronization
    - constant energy drain
- **Some of the presented protocols take significant advantage of WSN peculiarities like:**
  - small propagation delays
  - the ability to influence the node firmware to timestamp outgoing packets late, incoming packets early
- **More schemes exist....**

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

15th Lecture
13.12.2006