Software Infrastructures for Sensor Networks

Kay Römer et al
Distributed Systems
ETH Zürich
Switzerland

Previous Work

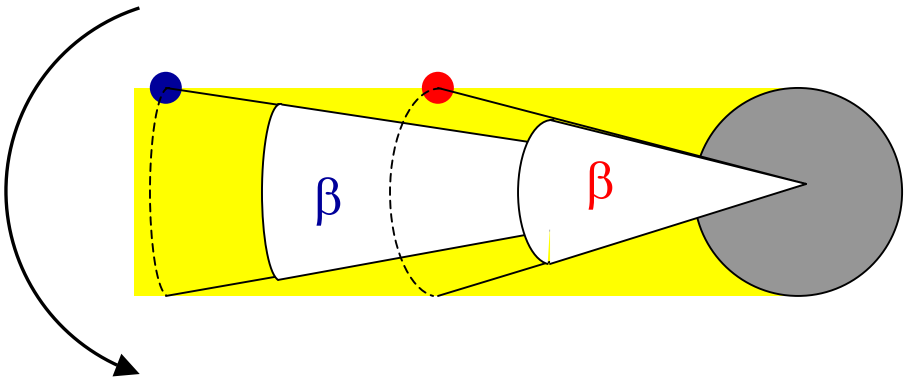
- Provision of key services for sensor networks
  - Node localization
  - Time synchronization
- Application prototyping
  - Object tracking
  - Product monitoring

Locating Smart Dust

- How to localize large populations of „Smart Dust"?
  - Tiny (mm^3) autonomous devices
  - Sensing, computing, wireless comm., power supply
- Key issues
  - Challenging device features (e.g., optical communication)
  - Energy efficiency
  - Scalability
  - Accuracy

Lighthouse Approach

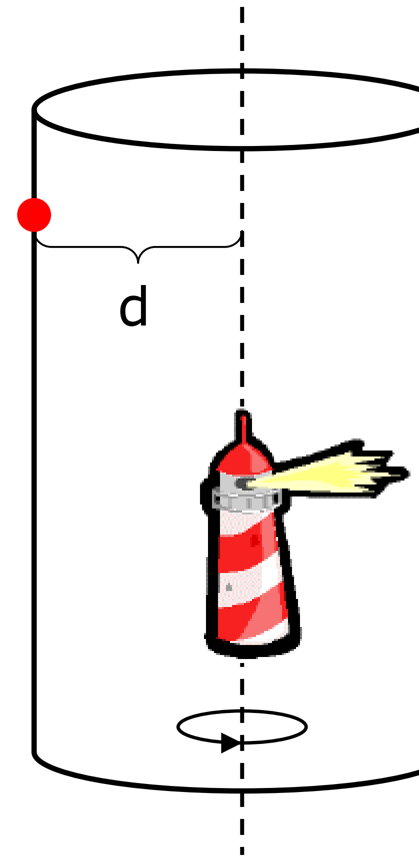
- Special lighthouse with parallel beam
  - Observer looks at lighthouse



- β depends on observers distance from lighthouse rotation axis!

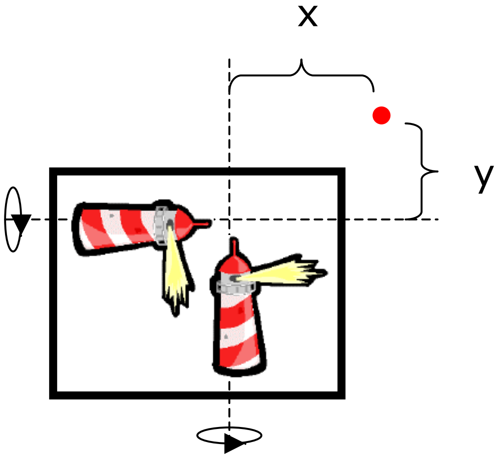
Lighthouse Approach

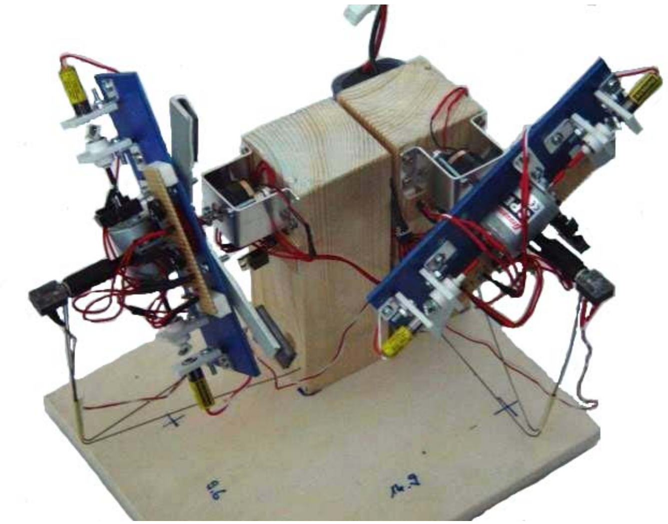
- We obtain distance to the lighthouse rotation axis!
- All observer locations with given d form the hull of a cylinder
- Localization approach
  - Multiple lighthouses
  - Compute intersection of cylinder hulls



Lighthouse Location System

- 2D: two lighthouses with perp. axes
  - Rotation axes define coordinate system
  - Distances from axes are 2D coordinates
  - Combine lighthouses into single device
- 3D: three lighthouses
  - Intersection of three cylinders



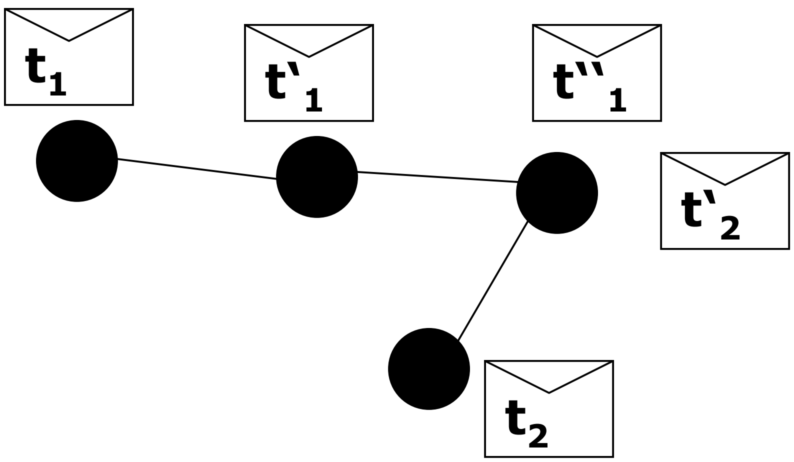


Time Sync for Sensor Nets

- Traditional network time sync
  - Sync all nodes, all of the time, at highest possible precision
  - Based on continuously synchronizing clocks
- Key issues
  - Energy efficiency
  - Scalability
  - Robustness (despite network dynamics)

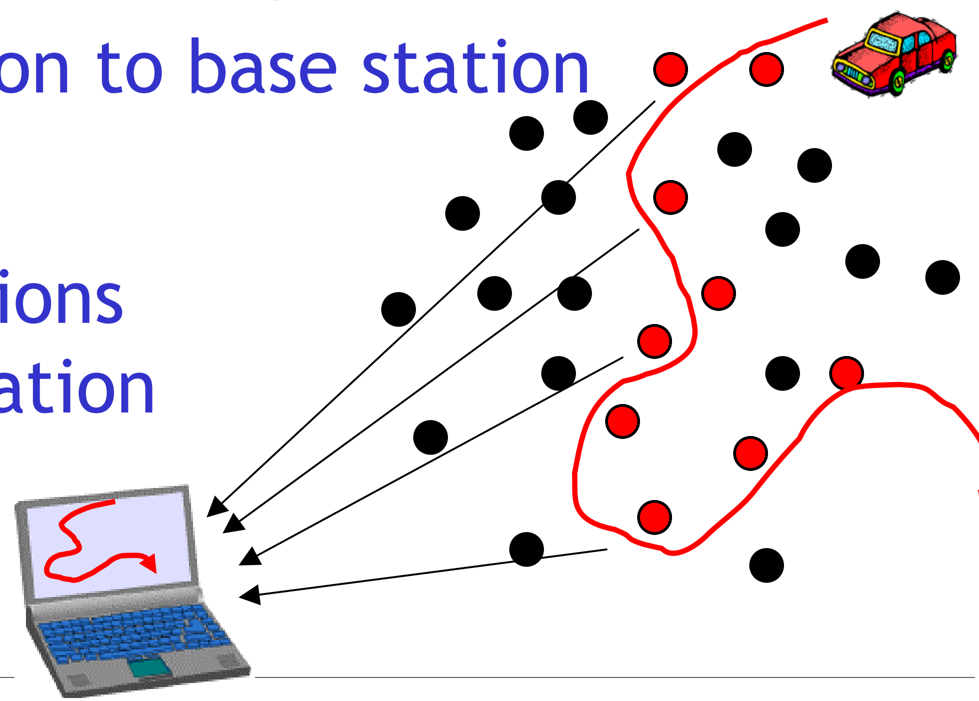
Timestamp Synchronization

- Synchronize clock readings (timestamps) instead of clocks
  - Sufficient for many applications
  - Can be done on demand
  - Can be piggybacked on data transfers



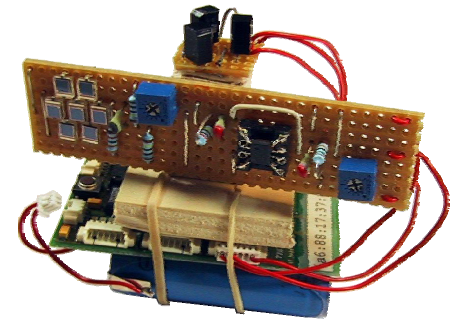
Tracking Application

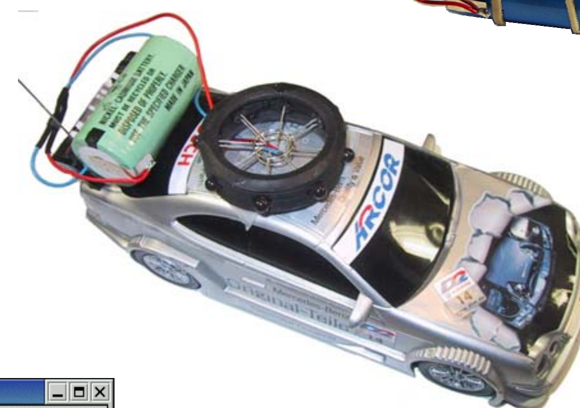
- Proof of concept for time sync and localization approaches
- Randomly deployed sensor nodes
  - Detect presence of target
  - Send notification to base station
- Base station
  - Fuses notifications using time/location
  - Displays track

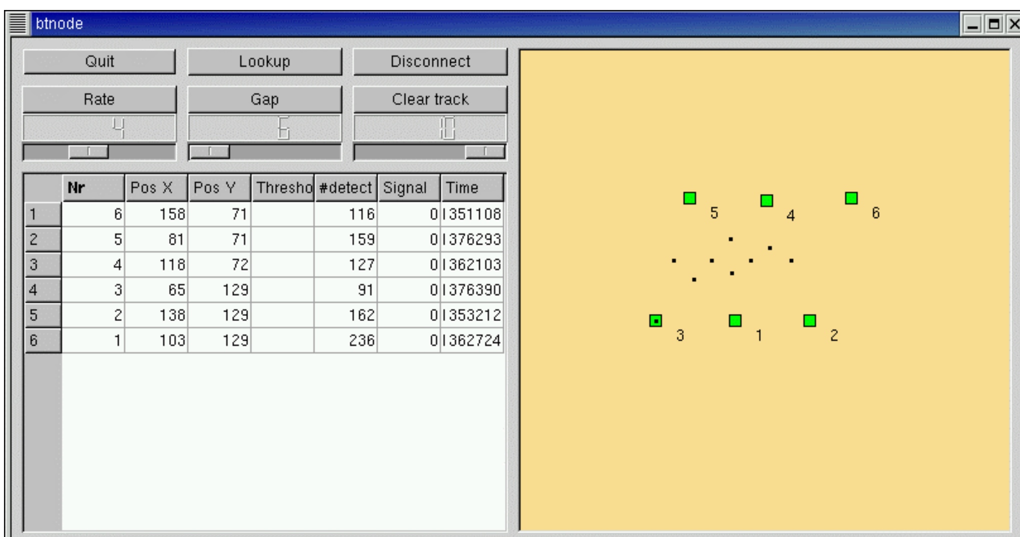


Prototype Implementation

- Car
  - Remote-controlled toy car
  - IR light emitter
- Sensor nodes
  - BTnodes
  - IR detector







Ongoing Work

- Programming sensor networks is a difficult task
  - Gap between problem-oriented task description and system-oriented programming of sensor networks
  - Requires expert knowledge in programming distributed embedded systems
  - Error-prone, debugging difficult, ...
- Goal: provision of high-level programming abstractions, tools, software infrastructures
  - Self-configuration
  - Target classification

Role-based Self-Configuration

- Many applications require heterogeneous node functions („roles")
  - Coverage: ACTIVE, STANDBY
  - In-network agg.: SOURCE, AGGREGATOR, SINK
- Assignment of roles to nodes may depend on
  - Hardware capabilities (sensors, memory, ...)
  - Other parameters (location, remaining energy)
  - Network neighborhood
- Framework for generic role assignment
  - Property directory
  - Role specification language
  - Distributed role assignment algorithm

Target Classification

- Common functionality:
  - What kind of vehicle?
  - Human or animal?
  - Friend or enemy?
- Framework for target classification
  - Allows specification of target properties
  - Color, size, weight, sound, ...