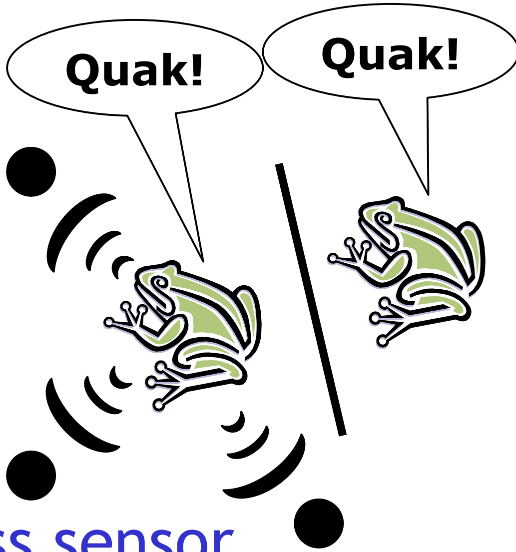
Locating Sensor Nodes in Time and Space

Kay Römer
ETH Zürich
Switzerland

Outline

- Why are time and location important in sensor networks?
- Why are they difficult to obtain?
- Our approaches
  - Localization
  - Time synchronization
- Application experience

Why Needed?



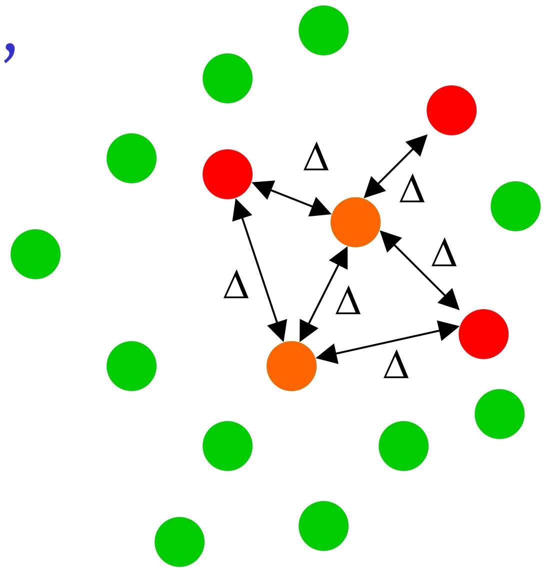
- Data Evaluation
  - Identify cause of real-world events
  - Separate distinct events
  - Fuse data from distributed sensors
- Addressing
  - Specify space-time regions to address sensor nodes, rather than ID-based addressing
- Distributed Coordination
  - Coordinate actions on distributed sensors
  - E.g., turn radio on/off
- Traditional uses

Why Difficult?

- Restricted size/cost/resources/energy
  - Precludes many traditional approaches and enabling technologies
- Network dynamics
  - Hardware failures, network partitions, obstructions, mobility, high/variable latency
- Scale of deployments
  - ensly deployed nodes
- untethered operation
  - ture, without manual configuration

Typical Approach

- Applies both to time sync and localization
- Few Anchors
  - Known time/location (via out-of-band mechanism)
  - >=1 for time, >=3 for location, but typically many more
  - „Well“ placed
- Other nodes
  - Measure offset Δ to nodes with known t/l
  - >=1 for time, >=3 for location
  - Infer own t/l



Potential Problems

- Accuracy
  - Distance from to anchors
  - Number of anchors
  - Placement of anchors
  - Accuracy of Δ measurements
- Infrastructure
  - Number and placement of anchors matters
  - Out-of-band mechanism for anchor synchronization
- Energy overhead
  - Proactive, always active
- Robustness
  - (Temporary) network partitions, ...

Solutions?

- Are there solutions to these problems?

The Lighthouse Location System for Smart Dust

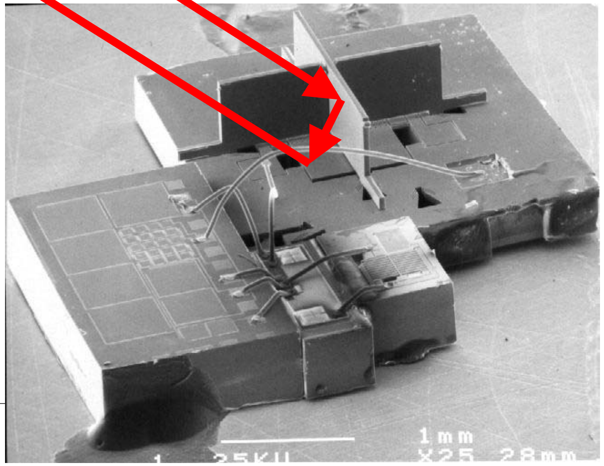
Locating Smart Dust

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

BST

Smart Dust Prototype

- Developed at UC Berkeley
- Avoid radio communication
  - Antennas larger than whole device
  - Transceiver power consumption
- Passive laser-based communication
  - Downlink: base station points modulated laser at dust particle
  - Uplink: dust modulates and reflects beam
  - Laser sweeps area of interest

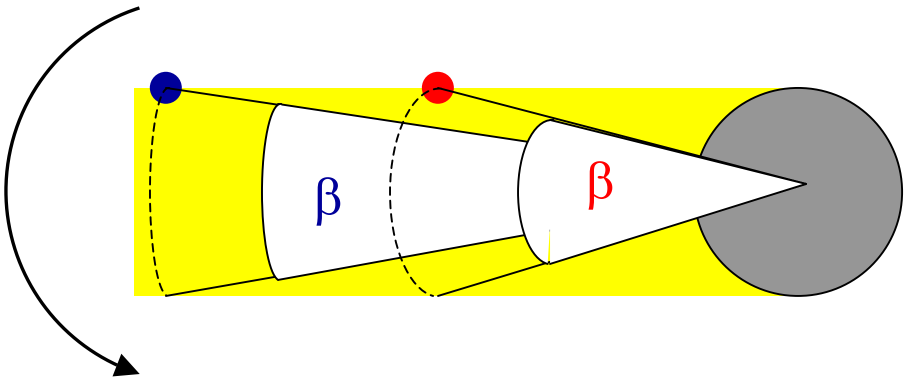


Lighthouse Approach

- Reuse dust node´s optical receiver for localization
- Infer location from laser light emitted by a (modified) base station
- Nodes do this autonomously
  - No communication with other nodes
  - No interaction with base station
  - „Passive observation“

Lighthouse Approach

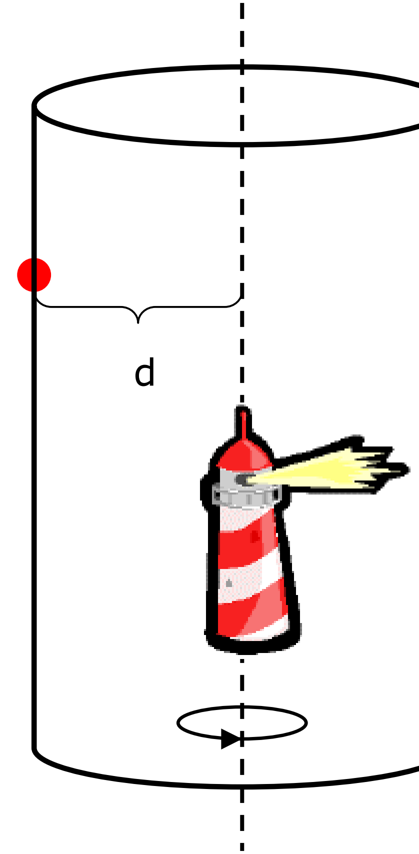
- Special Lighthouse with parallel beam
  - Observer looks at lighthouse



- β depends on observers distance from lighthouse rotation axis!

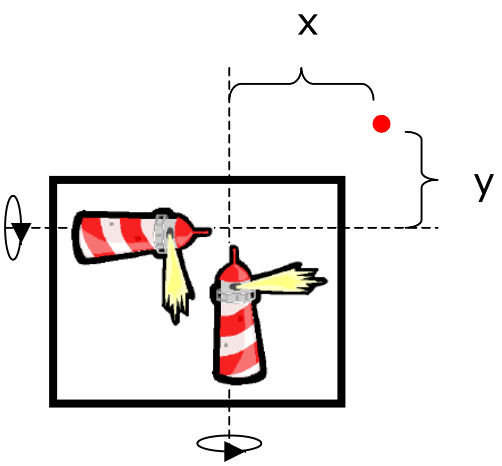
Lighthouse Approach

- We obtain distance to the lighthouse rotation axis!
- All observer locations with a given d form the hull of a cylinder



Location System

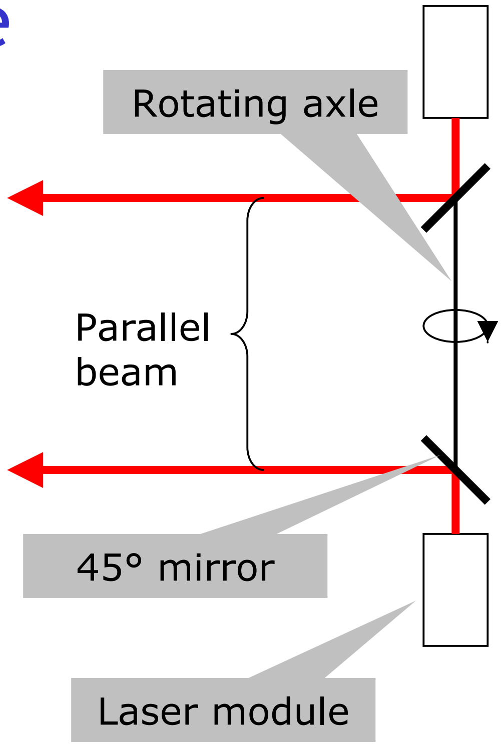
- 2D: two lighthouses with perp. axes
  - Rotation axes define coordinate system
  - Distances from axes are 2D coordinates
  - Combine lighthouses into single device



- 3D: three lighthouses
  - Intersection of three cylinders

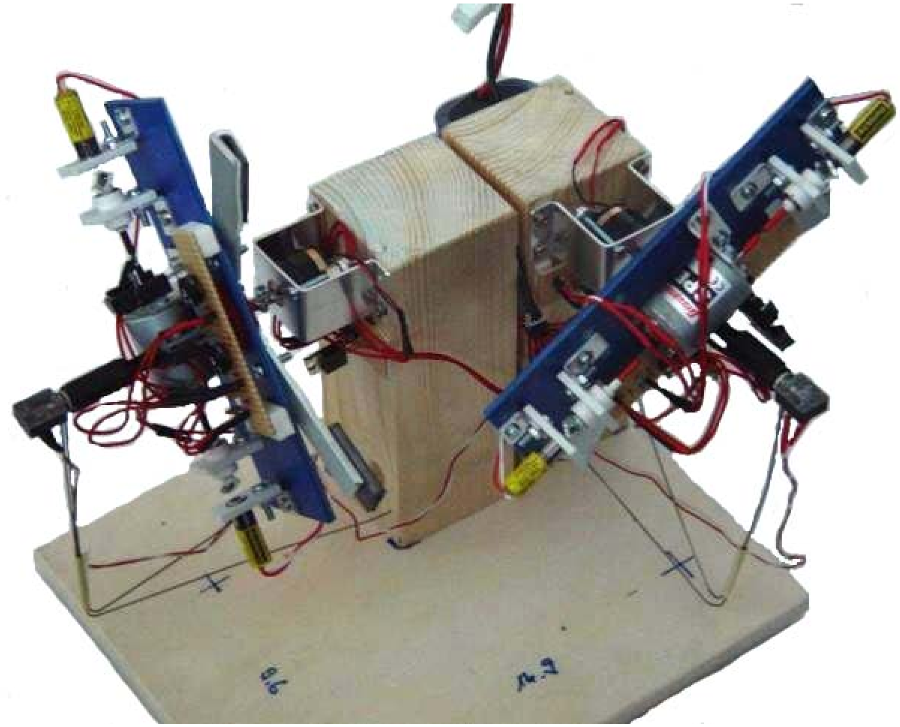
Lighthouse Implementation

- Beam generation
  - Two laser beams form outline of wide parallel beam
  - Rotating 45° mirrors
- Beam not parallel
  - More complex non-linear lighthouse model
  - Iterative solution
  - Lighthouse calibration
  - See MobiSys 03 paper for details



Lighthouse Prototype

- Based on two rotating laser beams
  - Two light „cones“
  - Virtual parallel beam, 12cm wide
  - 15000 rounds per minute (rpm), 250 Hz
- Mounted on rotating platform
  - 1 rpm



Accuracy

- Room-scale experiment (6m x 6m)
  - Mean error ~2% of distance
  - Standard deviation ~0.75% of distance
- Main sources of inaccuracies
  - Mechanical vibrations
  - Flutter of platform rotation
  - Beam/platform rotation speed
- Can be significantly improved with MEMS technology

Conclusion

- Scalable
  - No inter-node communication
  - Nodes autonomously compute own location
- Energy efficient
  - Nodes do not emit any signals, passive observation
- Fits the constraints of Smart Dust
  - No additional hardware on the nodes
  - Low computing, memory footprint
  - Single base station device
- Accuracy
  - Error within 2% of distance from basestation

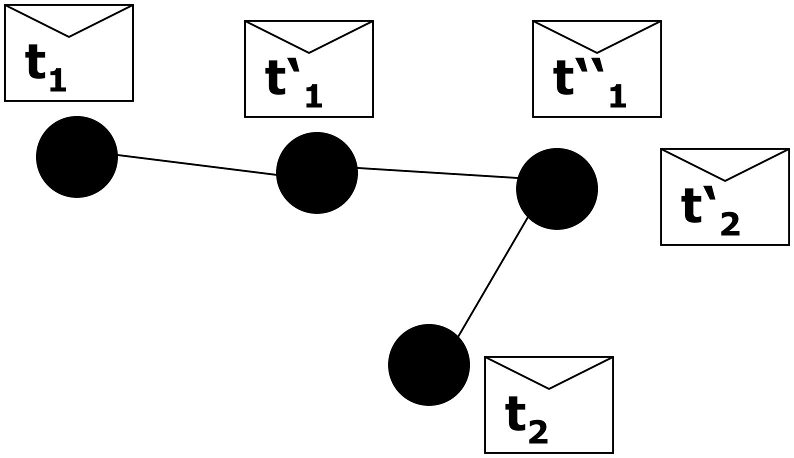
Timestamp Synchronization

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 we want to address
  - Energy efficiency
  - Scalability
  - Robustness

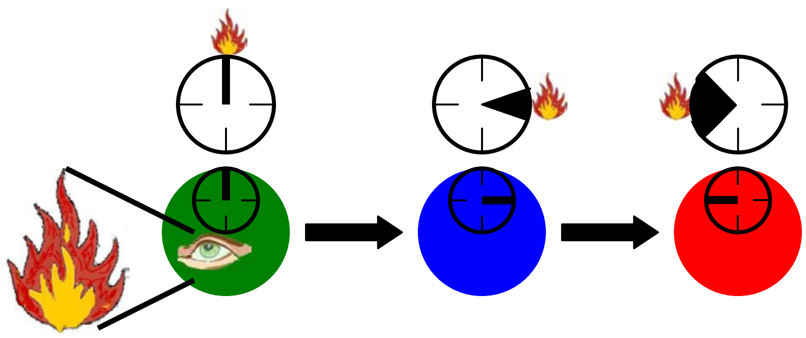
Basic Approach

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



Timestamp Synchronization

- Unsynchronized local clocks
- Messages carry timestamps
- Timestamps are transformed to receiver´s time upon message exchange



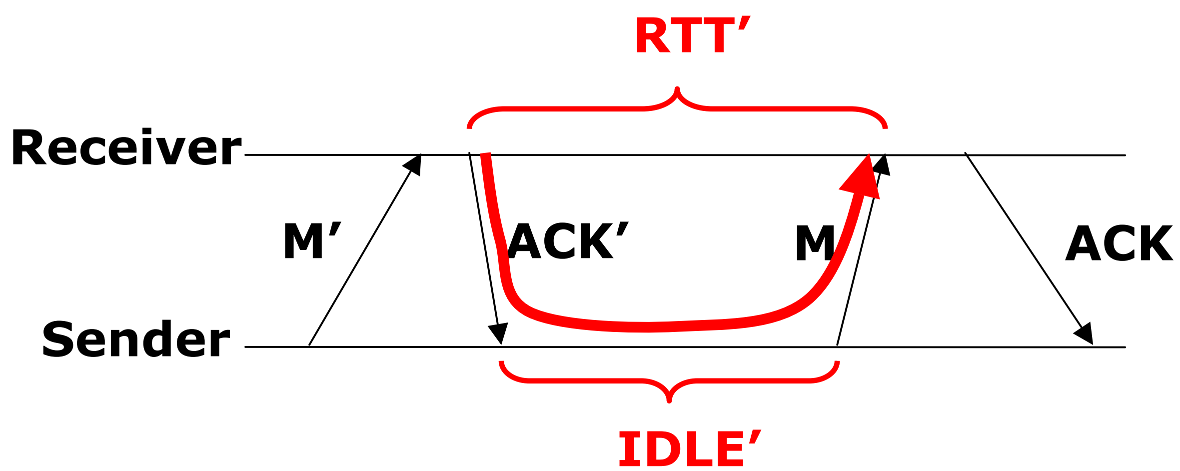
- Uncertainty intervals instead of time instants due to clock inaccuracies

Timestamp Transformation

- Determine age of time stamp and subtract from time of arrival
  - Age := storage time + transfer time
  - Storage time := $\Sigma\ t_{send}-t_{revc}$
  - Transfer time := Σ message delays

Message Delay

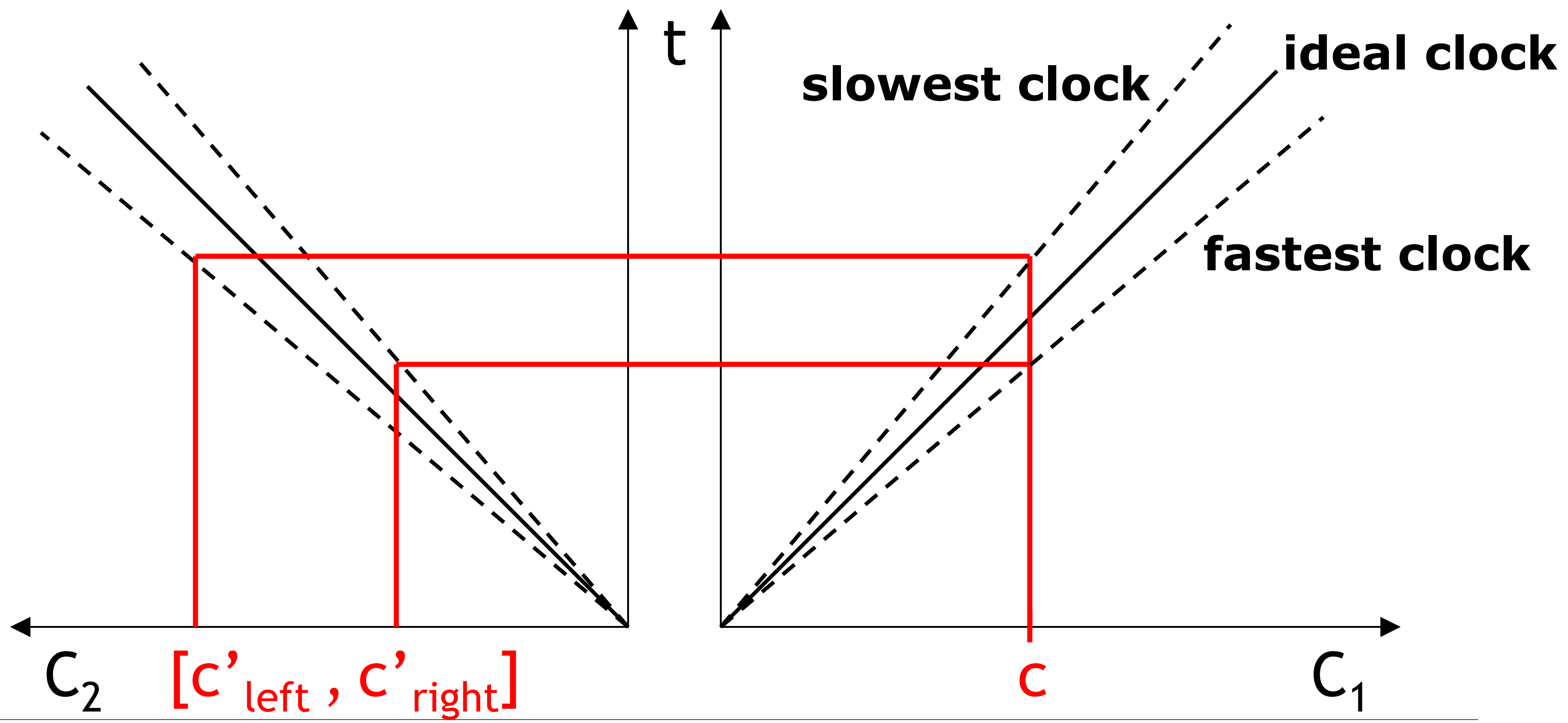
- Receiver needs to know message delay D for each message received from an *adjacent* node



- Sender knows: 0 < D < RTT - IDLE
- Receiver knows: 0 < D < RTT' - IDLE'

Time Transformation

- Each node i equipped with computer clock C_i which approximates real-time t
- Clocks with bounded drift



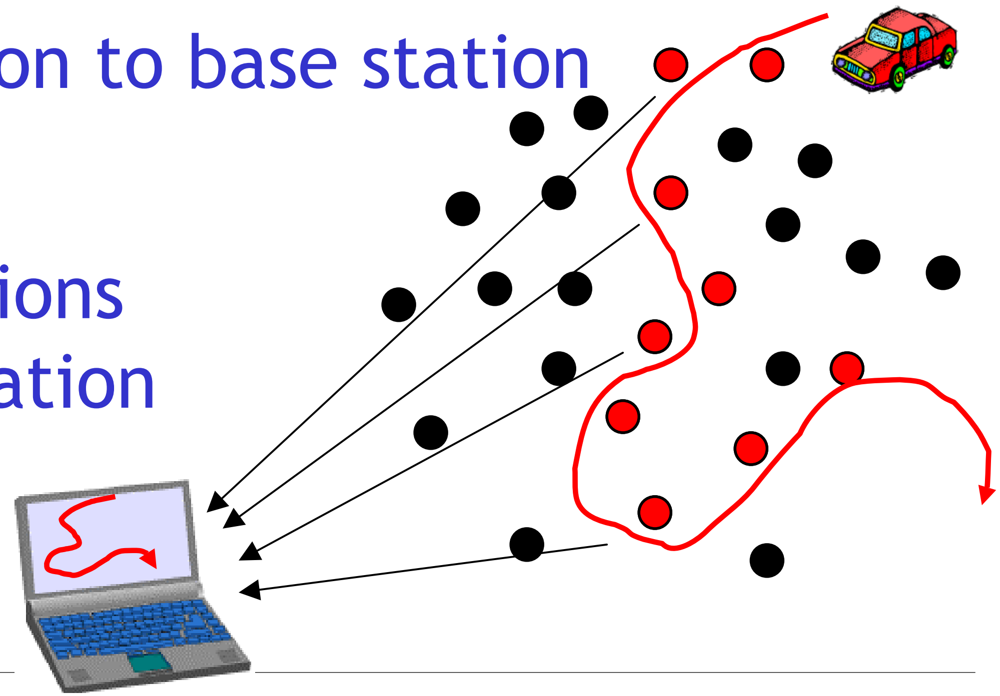
Conclusion

- Energy efficient
  - Only syncs where and when needed by the application
  - Can be piggybacked to existing message exchanges
  - Few additional message exchanges
- Scalable
  - Local interactions
- Robust
  - Works across temporary network partitions
- Accuracy: few milliseconds
  - 5 hops
  - 1000 seconds timestamp age
- See MobiHoc 01 paper for details

Application Experience

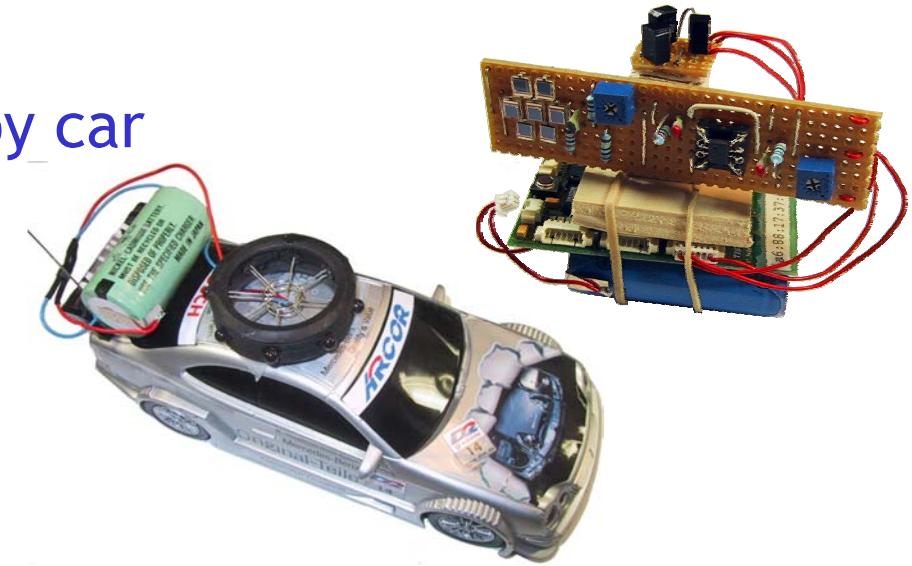
Tracking Application

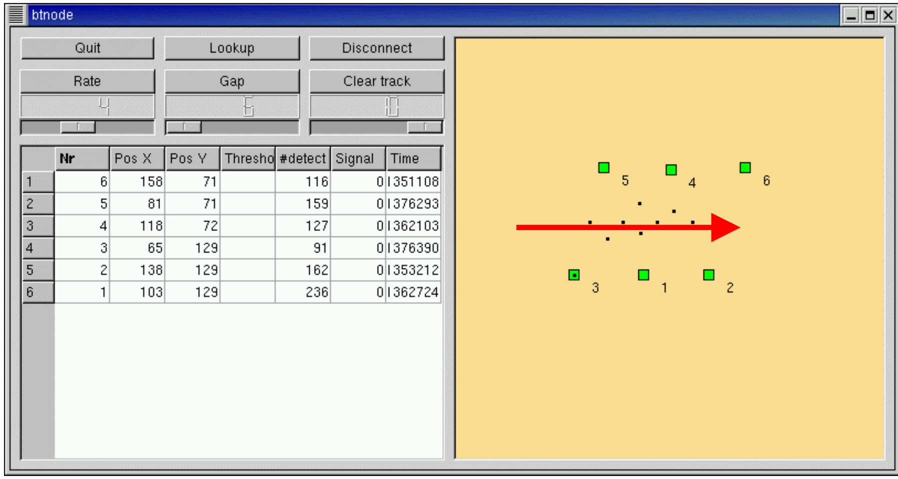
- Proof of concept for time sync and localization approaches
- Randomly deployed sensor nodes
  - Detect presence of target
  - Send notification to base station
- Base station
  - Fuses notifications using time/location
  - Displays track



Prototype Implementation

- Car
  - Remote-controlled toy car
  - IR light emitter
- Sensor nodes
  - BTnodes
  - IR detector
- Evaluation
  - Test setup: 6 nodes within 1m²
  - Average error < 12cm
  - Maximum error < 30cm
- See EWSN 04 paper for details





	Nr	Pos X	Pos Y	Thresho	#detect	Signal	Time
1	6	158	71		116	0	1351108
2	5	81	71		159	0	1376293
3	4	118	72		127	0	1362103
4	3	65	129		91	0	1376390
5	2	138	129		162	0	1353212
6	1	103	129		236	0	1362724