Distributed Intelligent Systems – W12:
An Introduction to (Static) Wireless Sensor Networks
Outline

• Motivating applications

• The Sensorscope project

• Tools used in this course

• Energy-saving design principles

• Examples of (distributed) intelligent algorithms for energy efficiency in wireless sensor networks
Motivating Applications
Motivation

- Micro-sensors, on-board processing, and wireless interfaces all feasible at very small scale
  - can monitor phenomena “up close”
- Will enable spatially and temporally dense environmental monitoring
- Will enable precise, real-time alarm triggering
- Embedded networked sensing will reveal previously unobservable phenomena

Adapted from D. Estrin, UCLA
Application 1 - *Permasense*

- What is measured:
  - rock temperature
  - rock resistivity
  - crack width
  - earth pressure
  - water pressure

*Pictures: courtesy of Permasense*
Application 1 - *Permasense*

- **Why:**
  
  “[…], gathering of environmental data that helps to understand the processes that connect climate change and rock fall in permafrost areas”

*Pictures: courtesy of Permasense*
Application 2 - GITEWS

German Indonesian Tsunami Early Warning System

• What is measured:
  – seismic events
  – water pressure

Pictures: courtesy of Deutsches GeoForschungsZentrum (GFZ)
Application 2 - GITEWS

• Why:

To detect seismic events which could cause a Tsunami. Detect a Tsunami and predict its propagation.

Pictures: courtesy of Deutsches GeoForschungsZentrum (GFZ)
Application 3 - Sensorscope

• What is measured:
  – temperature
  – humidity
  – precipitation
  – wind speed/direction
  – solar radiation
  – soil moisture

Pictures: courtesy of SwissExperiment
Application 3 - Sensorscope

• Why:
  Capture environmental events with high spatial density.

*Pictures: courtesy of SwissExperiment*
The SensorScope Project
Introduction

Temperature

Humidity

Light
Topology

?
Topology
Topology
Topology

GPRS
Topology

Pros
• Very simple!
• Essentially no restrictions in sensor locations

Cons
• The closest server access point may be quite far from the stations
• A long-range link may consume a lot of energy!
Topology

Power Consumption [mA]

0.5 3 50
Sensor MSP430 XE1205

(Microcontroller) (Short range radio, up to 2 km)
Topology

Power Consumption [mA]

Sensor: 0.5
MSP430: 3
XE1205: 50
GPRS: 700!

14x times the XE1205!
Topology

Assuming four AA batteries, 1.2 V, 2000 mAh

- Sensor: 167 days
- MSP430: 28 days
- Short range radio: 1.7 days
- Long range radio: 8 hours
Topology
Topology

Short range

GPRS

Sink
Topology

Friis law (power decay in air)

\[ L = \left( \frac{4\pi df}{c} \right)^2 \]

\[ P_L = 20 \log \left( \frac{4\pi d}{\lambda} \right) \]

Example: To transmit over 5 Km on 868 MHz we can use:

- One hop of 5 km: \( P_L = 106 \text{ dB} \)
- Two hops of 2.5 km: \( P_L = 99 \text{ dB} \)
- Five hops of 1 km: \( P_L = 92 \text{ dB} \)

Energy is the main issue !!!
Multi-hop WSNs

GPRS
Multi-hop WSNs

Pros

• Only one car battery in the network
• Extended spatial coverage of the network
• Multiple routes for stations to communicate with the sink
• Auto configurable network (robustness)

Cons

• Significantly more complicated
• Data rate is not increased
• Unable to use directional antennas
Multi-hop WSNs

Implementation:

• Neighborhood discovery
• Data routing
• Time synchronization
• Duty-cycling (radio management)

Pre-condition
Pre-condition
Neighborhood

Hello messages (Beacons) are one common method:
1. Node A sends a HELLO message to its neighbors (B, C, and D).
3. Node B sends a HELLO message to its neighbors (A, C, and D).
4. …
Hello messages (Beacons) are one common method:
1. Node A sends a HELLO message to its neighbors (B, C, and D).
3. Node B sends a HELLO message to its neighbors (A, C, and D).
4. …
Neighborhood

Hello messages (beacons) are one common method:
1. Node A sends a HELLO message to its neighbors (B, C, and D).
3. Node B sends a HELLO message to its neighbors (A, C, and D).
4. ...
Neighborhood

Hello messages (beacons) are one common method:
1. Node A sends a HELLO message to its neighbors (B, C, and D).
3. Node B sends a HELLO message to its neighbors (A, C, and D).
4. …
Neighborhood

What information do we need about our neighbors?

- Distance to sink
- Last time heard
- Link quality
Neighborhood

Node E’s neighborhood table

<table>
<thead>
<tr>
<th>Id</th>
<th>Age</th>
<th>Distance</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>2 min</td>
<td>1 hop</td>
<td>87%</td>
</tr>
<tr>
<td>C</td>
<td>2 min</td>
<td>1 hop</td>
<td>98%</td>
</tr>
<tr>
<td>F</td>
<td>4 min</td>
<td>2 hops</td>
<td>74%</td>
</tr>
<tr>
<td>G</td>
<td>1 min</td>
<td>2 hops</td>
<td>93%</td>
</tr>
</tbody>
</table>

A few remarks:

• Only the distance to the sink is stored.
• Neighborhood discovery can’t be done only once!
• We need to estimate link qualities!
Variations of simple schema:

- Each node sends $X$ beacons per minute.
- Number of beacons received per minute are stored.
- Quality is estimated over the past $Y$ minutes by counting losses.

Example ($X = 10; Y = 4$):

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$t-4$</td>
<td>$t-3$</td>
<td>$t-2$</td>
<td>$t-1$</td>
<td>Quality</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>0.8</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>0.71</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>6</td>
<td>5</td>
<td>0.64</td>
</tr>
</tbody>
</table>
Time Synchronization

Weather conditions, especially temperature and humidity, may have a significant effect on hardware.

Crystal oscillators are highly impacted by temperature!
Time Synchronization

![Graph showing air temperature and time drift for Indoor, Outdoor, and Freezer environments over a period of 8 days.](Image)
Time Synchronization

Nodes need to know the time to:
• Timestamp packets
• Synchronize actions (e.g., taking samples, transmitting data)

How do we get time:
• Fully decentralized: Every node gets the time itself
• Partially centralized: Time is propagated from reference nodes
Time Synchronization

Every node gets the time:

• Atomic clock receivers:
  • Cheap (both energy and $)
  • Complexity
  • Limited coverage

• GPS:
  • Coverage
  • Complexity
  • High cost (energy and $)

• GPRS: same as GPS with less coverage

What about a partially centralized approach?
Time Synchronization

For instance, the sink serve as time reference node
Visualizing Filter "Public Stations"

Local time: 2:59 (GMT+1)

Station 1049

Meteorological data
- Davis Anemometer Direction: West (288.8°)
- Davis Anemometer Speed: 2.7 m/s
- SHT75 Humidity: 65.5%
- SHT75 Temperature: -13.6°C

Health status
- Battery - External: Not connected
- Battery - Internal: 5.4 V
- CPU - Temperature: -17.9°C
- CPU - Voltage: 3 V
- MMC Card Free Space: 964 MB

Plot recent data
- Temperature graph: Current value -5.38°C
Hardware and Software Modules used in this Course
MICA mote family

- designed in EECS at UCBerkeley
- manufactured/marketed by Crossbow
  - several thousand produced
  - used by several hundred research groups
  - about CHF 250/piece
- variety of available sensors
MICAz

- **Atmel ATmega128L**
  - 8 bit microprocessor, ~8MHz
  - 128kB program memory, 4kB SRAM
  - 512kB external flash (data logger)

- **Chipcon CC2420**
  - Respect 802.15.4 at the physical/MAC layer and therefore can support Zigbee-compliant stacks

- **2 AA batteries**
  - about 5 days active (15-20 mA)
  - about 20 years sleeping (15-20 µA)
Sensor board

- MTS 300 CA
  - Light (Clairex CL94L)
  - Temp (Panasonic ERT-J1VR103J)
  - Acoustic (WM-62A Microphone)
  - Sounder (4 kHz Resonator)
TinyOS

- Minimal OS designed for Sensor Networks
- Event-driven execution
- Programming language: nesC (C-like syntax but supports TinyOS concurrency model)
- Widespread usage on motes
  - MICA (ATmega128L)
  - TELOS (TI MSP430)
- Provided simulator: TosSim
802.15.4 / Zigbee

- Emerging standard for low-power wireless monitoring and control
  - 2.4 GHz ISM band (84 channels), 250 kbps data rate

- Chipcon/Ember CC2420: Single-chip transceiver
  - 1.8V supply
    - 19.7 mA receiving
    - 17.4 mA transmitting
  - Easy to integrate: Open source drivers
  - O-QPSK modulation; “plays nice” with 802.11 and Bluetooth
Comparison to other standards

Complexity/power/cost

Data rate

- **802.11b**: 11 Mbps, 720 kbps
- **802.11g**: 54 Mbps
- **802.11a**:
- **Bluetooth**: 250 kbps
- **802.15.4 Zigbee**: 38.4 kbps
- **CC1000**: 38.4 kbps
Basic Principles for Energy-Saving Design in Static Wireless Sensor Networks
Energy in WSN

– Energy saving is a crucial driver for the design of WSN
– Sensing data are typically only collected for a particular application and rarely used to control node actions: WSN are typically data-agnostic!
– Large cost of communication relative to computation

<table>
<thead>
<tr>
<th></th>
<th>1999 (Bluetooth Technology)</th>
<th>2004</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>(150nJ/bit)</td>
<td>(5nJ/bit)</td>
</tr>
<tr>
<td></td>
<td>1.5mW*</td>
<td>50uW</td>
</tr>
<tr>
<td>Computation</td>
<td>~ 190 MOPS</td>
<td>(5pJ/OP)</td>
</tr>
</tbody>
</table>

Assume: 10kbit/sec. Radio, 10 m range.

Source: ISI & DARPA PAC/C Program
Generalization: Friis Laws

- **Basic Friis law (open environment)**

\[
\frac{P_r}{P_t} = G_t G_r \left( \frac{\lambda}{4\pi R} \right)^2
\]

- **Modified Friis law (cluttered, urban environment)**

\[
\frac{P_r}{P_t} = G_t G_r \left( \frac{\lambda}{4\pi} \right)^2 \left( \frac{1}{R} \right)^n
\]

\[n \text{ between 2 and 5!}\]

\[f = \frac{c}{\lambda}!\]
Communication Power Budget: Not only Transmission

• In general transceiver power consumption dominated by listening (radio on)
  – ChipCon CC2420: 19.7 mA in receiving, 17.4 mA transmitting @ 0 dBm (2.4 GHz) [Mica-Z]
  – SemTech SX 1211: 3 mA in receiving, 25 mA @ +10 dBm in transmitting (900 MHz) [SensorScope]

• Synchronization is key

• Low power listening protocols are key for saving power
Efficient Monitoring in Wireless Sensor Networks
WSN and DIS - Motivation

**A few key questions:**

– Can additional *in-node/in-network intelligence* help in saving energy?

– Does it make sense to have WSN *data-aware* (as opposed to the canonical data-agnostic paradigm)?

– Is there a way to make a *stronger overlap between WSN and DIS*, in particular more closed-loop control at the node level, more distributed control at the network level?
Motivation

Minimize energy resources used – Maximize field accuracy obtained

- Sample as little as possible
- Communicate as little as possible

- In space
- In time

- # of messages
- Comm. radius
Opportunities for Spatiotemporal Suppression in Environmental Data

- Ambient temperature
- Surface temperature
- Relative humidity

Graphs showing correlation coefficients and temperature over distance and time.
Clustering and Threshold-Based Pruning
Case study: Estimating an Acoustic Field

- 4^n sensing cells
- 4^n sensor nodes
- 1 static sound source

Performance metric:
1. MSE (data quality)
2. A (number active nodes)

Objective function:
\( f_{obj}(\text{MSE}, A) \)
Multiple roles → layers

Hierarchy:
  - Bottom-up measurements
  - Top-down control

Hierarchical Topology – Quadtree

**Single node level**

- Messaging protocol dependant on:
  - *sender-ID*
  - *sender-layer*

- Channels given by quadtree

![Quadtree diagram](image.png)
Distributed Control – State Machine

- Each node is assigned a specific $L_{k_{max}}$ by architectural design
- **Layer increment**: if all child nodes processed
- **Idle node**: if pruned by clusterhead
- **Data**: clusterhead replaces pruned children
An Intuitive Illustration

Naïve Sampling  
Threshold-based Sampling

(a)  
(b)  

(c)  
(d)
Experimental Setup

The e-puck robot:
- Trinaural microphone array (28.8 Hz)
- Short range communication
  - subset of 802.15.4 Zigbee
  - comm. range: 10cm - 5m

Setup:
- 1.5m x 1.5m arena
- 16 e-puck nodes (static sensor stations)
- 1 static sound source (white noise, const. intensity)
Experimental Results

Number of active nodes

MSE

Setup:
• 10 runs, variable source placements / threshold
• 12 thresholds, with $s$ in $[0..12'000]$
• Model with $t_x = 0.3$
Spatiotemporal Suppression of Data Reporting
Temporal suppression
  • Has my value changed recently?

Spatial suppression
  • Are my neighbors reporting similar measurements?
Temporal Suppression

Ex: Barebones temporal suppression (field change in gray)
Temporal Suppression

Ex: Barebones temporal suppression (reporting node in red)
Temporal Suppression

Ex: Barebones temporal suppression
Temporal Suppression

Ex: Barebones temporal suppression
Efficient Monitoring

Monitor edge constraints instead of individual nodes
Efficient Monitoring

Monitor edge constraints instead of individual nodes
Efficient Monitoring

Monitor edge constraints instead of individual nodes
Efficient Monitoring

Monitor edge constraints instead of individual nodes
Constraint Chaining

- Suppression-based algorithm: Constraint Chaining (Conch) [1]
- Monitor edge constraints instead of sensor values
- Historical data to identify performant edges

Testbed
In-Network Power Monitoring

Sensors:
- Battery voltage
- Solar voltage
- Incoming solar current
- Datalogger current
- Power board current
- Sensor chain current
Results with Basic CONCH

• Four weeks on the outdoor testbed
• Limited adaptivity to network changes
  • Replanning costs $O(|E|) + \text{GPRS}$
• Unstable network led to most nodes reporting directly to the sink
• 45% of messages suppressed while algorithm was functional
Results with Advanced CONCH

- Autoregressive model for more compressed comparison of differences
- AR-Conch: 57% suppression rate

- Autoregressive model + in-network distributed implementation for enhanced adaptive behavior
- AR-DConch: 64% suppression rate
Concrete Energy Savings

Not significant since:
• Radio on for several seconds (overhearing)
• Transmission takes 6 ms max
• Two minutes idle at ~11.8mW

Energy savings given fixed 50% suppression rate (calibrated simulation in TOSSIM)

Our settings
Conclusion
Take Home Messages

- WSNs represent a very promising technology for a number of applications.
- Environmental data are (usually) highly redundant.
- Embedded intelligence at the node/network level has the potential to remove that redundancy and save energy.
- AI techniques have been studied in simulation but they are difficult to bring to real systems because both partially predictable dynamic environmental processes and dynamic network conditions.
- Network stack may limit potential gains in energy saving of an intelligent algorithm; it is often a question of robustness versus efficiency.
Additional Literature – Week 12

Books

Papers

Pointers
• Sensorscope: http://sensorscope.epfl.ch/
• Permasense: http://www.permasense.ch/
• GITEWS: http://www.gitews.de
• CENS: https://en.wikipedia.org/wiki/Center_for_EMBEDDED_NETWORK_Sensing
• WSN: https://en.wikipedia.org/wiki/Wireless_sensor_network
• TinyOS: https://en.wikipedia.org/wiki/TinyOS