Distributed Intelligent Systems – W07: An Introduction to Wireless Sensor Networks from a Distributed Intelligent Systems Perspective
Outline

• Motivating applications
• The Sensorscope project
• Tools used in this course
  – Mica-z
  – 802.15.4 radio for e-puck robots
  – Webots extensions
• Wireless Sensor networks vs. Distributed Intelligent Systems
• Collective decisions and discrete consensus algorithms
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 meteorological 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

Sensor MSP430 XE1205 GPRS

Power Consumption [mA]

0.5 3 50

14x times the XE1205!

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

Power Consumption [mA]

0.5  3  50  700!
Sensor  MSP430  XE1205  GPRS

14x times the XE1205!
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

Sink

GPRS
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: \( P_L = 106 \) dB
• Two hops of 2.5 km: \( P_L = 99 \) dB
• Five hops of 1 km: \( P_L = 92 \) 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)
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. …
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

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

<table>
<thead>
<tr>
<th>$t-4$</th>
<th>$t-3$</th>
<th>$t-2$</th>
<th>$t-1$</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<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 conditions over 8 days.](image-url)
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)
Hardware and Software Modules used in this Course
MICA mote family

• designed in EECS at UC Berkeley
• 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)

• TinyOS
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

- **802.11b**: 11 Mbps
- **802.11a**: 54 Mbps
- **802.15.4 Zigbee**: 720 kbps
- **CC1000**: 250 kbps
- **Bluetooth**: 38.4 kbps

Data rate
The epuck 802.15.4 Radio

- Custom module designed specifically for **short range**
- Software controllable (~5cm-5m)
- Radio stack implemented in TinyOS (not fully ZigBee compliant)
- Interoperable with MICAz, etc.

[Cianci et al, SAB-SRW06]
Communication Plug-In for Webots

Original Plug-in
- OmNET++
- OSI framework
- Custom Layers
  - 802.15.4
  - ZigBee
- Physical communication model:
  - semi-radial disk with noise
  - channel intensity fading
- Calibrated with real hardware

New Plug-in
- NS3 (more promising framework)
- 802.11p, uncalibrated
- [Llaster et al. 2017]
Wireless Sensor Networks vs. Distributed Intelligent Systems
WSN and DIS

Wireless sensor networks:
- are spatially distributed systems
- exploit wireless networking as main inter-node interaction channel
- typically consist of static, resource-constrained nodes
- energy saving is a crucial driver for the design of WSN
- have nodes which can sense, act (typically no physical movement), compute and communicate in an unattended mode

Are WSN a special class of Distributed Intelligent Systems?
WSN and DIS

The potential is there but currently we observe:

- **Limited embedded intelligence/adaptation:**
  - sensing data are typically only collected for a particular application and rarely used to control node actions: WSN are typically data agnostic!
  - no emphasis on local real-time perception-to-action loop
  - activity pattern (sensing, computing, networking) are typically a priori scheduled
  - static nodes face lower unpredictability than mobile ones

- **Limited control distributedness**
  - the fact that WSN are spatially distributed does not necessarily mean distributed control: the existence of a sink allows for centralized control which in turn often promote energy saving
WSN vs. Networked Multi-Robot Systems

Networking is common, sensor nodes = mobile robots without self-locomotion capabilities or mobile robots = sensor nodes with self-locomotion capabilities. So minimal difference? Not really …

• Mobility changes completely the picture of the problem: more unpredictability, noise, … .
• Self-locomotion even more so: real-time control loop at the node level + energy budget breakdown radically different
• Typically different system performance metrics and therefore different objectives pursued at design stage
Collective Decisions using WSN

• Collective decisions can be taken as benchmarking framework for testing distributed intelligent algorithms

• They do not involve necessarily mobility; interactions can happen via radio

• Discrete consensus algorithms (e.g., voting scheme) can be deployed on a WSN for reaching a collective decision
Collective Decisions
Understanding Collective Decisions

- Local rules and appropriate amplification and/or coordination mechanisms can lead to collective decisions
- **Modeling** to understand the underlying mechanisms and generate ideas for artificial systems

**Individual behaviors and local interactions**

**Ideas for artificial systems**

**Global structures and collective decisions**
Example 1: Selecting a Path (W1)

(Nest) ———— (Food source)

$L = 14 \text{ cm}$

Choice occurs randomly

(Deneubourg et al., 1990)
Example 2: Selecting a Food Source (W1)
Example 3: Selecting a Shelter

- **Leurre**: European project focusing on mixed insect-robot societies ([http://leurre.ulb.ac.be](http://leurre.ulb.ac.be))
- A simple decision-making scenario: 1 arena, 2 shelters
- Shelters of the same and different darkness
- Groups of pure cockroaches (16), mixed robot+cockroaches (12+4)
- Infiltration using chemical camouflage and statistical behavioral model
- More in week 14

[Halloy et al., *Science*, Nov. 2007]
Example 4: Selecting a Direction

Converging on the direction of rotation (clockwise or anticlockwise):
• 11 Alice I robots
• infrared-based local communication
• Idea: G. Theraulaz (and A. Martinoli); implementation: G. Caprari, W. Agassounon
Set-up and Discrete Consensus Algorithm

- 10 robots execute wall-following behavior (CW or CCW, initially random)
- announce their current direction on the radio channel

- # of votes not constant
- probabilistic decision
- communication range affects time to convergence.

[Cianci et al, SAB-SRW 06]
Some Results

- Larger communication ranges (neighborhoods) yield faster convergence

- Smaller communication ranges suffer more from partial perception; the majority near any given node may not reflect the majority in the system at a specific time

[Cianci et al, SAB-SRW 06]
Alternative Scenario: Networking
Sensor Nodes and Robots

- A similar experiment was performed in a classroom laboratory exercise.
- Each pair of students has a robot in an isolated arena.
- Near one side of the arena, the robot is able to communicate with a node in a sensor network, which shares the information with other nodes in the network, so that they may inform their corresponding robots.
- Convergence to a collective decision was also reached in this case (though a full suite of systematic experiments were not performed).

[Cianci et al, SAB-SRW 06]
Example 5: Assessing Acoustic Events

- In some situations, event detection may trigger a costly process (i.e. human intervention, fire brigade, etc...)
- A simple consensus mechanism may help limit false positives (e.g., require $k$ nodes to agree on detection before reporting)

Example with $k=2$

[Cianci et al., ICRA 2008]
Set-up

- 1.5 x 1.5m tabletop arena
- 16 sensor nodes (static e-pucks)
- 1 sound source (mobile e-puck)
- Sound and network plug-ins in Webots

Physical

Simulation (Webots)
Metrics and Experiments

- Events characterized by two intensities are presented to the network:
  - Events with targeted intensity $I_e$
  - Events with undesirable intensity $I_u$ (50%, 75%, 95% of $I_e$)
- Each experiment involves 100 events with the targeted intensity and 100 with the undesirable intensity
- Results report performance over 20 runs of the same experiment
- Performance metric:

$$M_E(\alpha, \beta) = \alpha \frac{E_{det}}{E_{tot}} + \beta \left( 1 - \frac{E_{fp}}{\max(E_{fp}, E_{tot})} \right)$$

$E_{det}$: the number of targeted events reported
$E_{tot}$: the total number of targeted events presented
$E_{fp}$: the number of false positives reported
Metrics

\[
M_E(0.5, 0.5) \\
\text{Composite}
\]

\[
M_E(1, 0) \\
\text{Detected events}
\]

\[
M_E(0, 1) \\
\text{False positives}
\]

Note: focus on “Physical System” (real robots) and “Module-based” (Webots)

[Cianci et al., ICRA 2008]
Conclusion
Take Home Messages

• WSNs represent a very promising technology for a number of applications

• Commonalities and synergies between distributed, networked multi-robot systems and WSNs are appearing but their potential need still to be further investigated and formalized

• Collective decisions represent interesting benchmarks for testing distributed consensus algorithms on WSNs

• Various forms of consensus algorithms have been considered; some of them involve locally and globally discrete decisions, other locally and globally continuous, and other again a combination of both discrete and continuous components at local and global level
Additional Literature – Week 7

- Permasense [http://www.permasense.ch](http://www.permasense.ch)
- GITEWS – the German Indonesian Tsunami Early Warning System [http://www.gitews.de](http://www.gitews.de)
- Sensorscope [http://www.sensorscope.ch/](http://www.sensorscope.ch/)
- Course list: [http://www-net.cs.umass.edu/cs791_sensornets/additional_resources.htm](http://www-net.cs.umass.edu/cs791_sensornets/additional_resources.htm)
- TinyOS: [https://github.com/tinyos](https://github.com/tinyos)
- Smart Dust Project [http://robotics.eecs.berkeley.edu/~pister/SmartDust/](http://robotics.eecs.berkeley.edu/~pister/SmartDust/)
- UCLA Center for Embedded Networking Center [http://auvac.org/people-organizations/view/386](http://auvac.org/people-organizations/view/386)
- NCCR-MICS at EPFL and other Swiss institutions [http://www.mics.org](http://www.mics.org)