Course “Algorithmic Foundations of Sensor Networks”
Lecture 1: Introduction / Data propagation algorithms I

Sotiris Nikoletseas
Department of Computer Engineering and Informatics
University of Patras, Greece

Spring Semester 2010-2011
Summary of this Lecture

• Introduction to Wireless Sensor Networks (characteristics, applications, critical challenges, algorithmic properties)
• Thematic structure of the 2010-2011 Course
• The problem of data propagation
• Two protocols for data propagation:
  • Directed Diffusion
  • LEACH
What is a Wireless Sensor Network?

- very large number of tiny “smart” sensors
- severe limitations
- wireless communication
- densely / randomly deployed in an area
- self-organization
- co-operation
- locality

An “ad-hoc” wireless network for:
- sensing crucial events
- data propagation
A smart dust “cloud”

Control Center

Sensor field

Sensor nodes
“Smart” Sensor Nodes

- very small size
- operate on a small battery
- multifunctional sensing
- wireless communication
- limited computing power / limited memory
- low cost
## TPR2420 Specifications (TelosB motes)

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dimensions</strong></td>
<td>6.5cm × 3.1cm × 2cm</td>
</tr>
<tr>
<td><strong>CPU</strong></td>
<td>8 MHz Texas Instruments MSP430</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>10KB RAM, 48KB Program flash, 16KB EEPROM, 1MB flash</td>
</tr>
<tr>
<td><strong>Power Supply</strong></td>
<td>2X AA batteries</td>
</tr>
<tr>
<td><strong>Processor Current Draw</strong></td>
<td>1.8 mA (active current)</td>
</tr>
<tr>
<td></td>
<td>&lt; 5.1 µA (sleep mode)</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>USB</td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td>Wireless 250 Kbps at 2.4GHz (802.15.4)</td>
</tr>
<tr>
<td></td>
<td>Radio range depends on configuration and location</td>
</tr>
<tr>
<td><strong>Integrated Sensors</strong></td>
<td>Visible Light and IR, Humidity, Temperature</td>
</tr>
</tbody>
</table>
## TinyOS Key Facts

<table>
<thead>
<tr>
<th>Current Version</th>
<th>TinyOS 2.0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>Over 500 research groups and companies</td>
</tr>
<tr>
<td>Software Footprint</td>
<td>3.4 Kb</td>
</tr>
<tr>
<td>Architecture</td>
<td>Component-based</td>
</tr>
<tr>
<td>Hardware Supported</td>
<td>8 different mote types</td>
</tr>
<tr>
<td>Peak load CPU Usage</td>
<td>&lt; 50 %</td>
</tr>
<tr>
<td>Prog. Language Used</td>
<td>NesC</td>
</tr>
<tr>
<td>Components provided</td>
<td>Sensor Interface</td>
</tr>
<tr>
<td></td>
<td>Basic Multihop Routing</td>
</tr>
<tr>
<td></td>
<td>Radio Communication</td>
</tr>
</tbody>
</table>
Examples of Motes

Mica2

- Atmel® ATMega128
- External power connector
- MMCX connector (female)
- 51-pin Hirose connector (male)
- On/Off Switch

TelosB / Tmote Sky
In traditional sensors:

- sensors perform only sensing
- sensor positions are carefully engineered
- deployed far from the area of interest
- data processing only at some central nodes
- larger in size
- constant power supply
• “Smart Dust" (Berkeley)
  • primary goal: a few $mm^3$ size
  • laser communication
  • motes development
  • TinyOS

(http://robotics.eecs.berkeley.edu/~pister/SmartDust/)
• WINS ("Wireless Intergrated Network Sensors") (UCLA)
  • focus: RF communication
  • development of low power MEMS

(http://www.janet.ucla.edu/WINS/)

• “Ultralow Power Wireless Sensor” (MIT)
  • primary goal: extremely low power consumption
  • power levels: 10 $\mu$W to 10 mW
  • RF communication

(http://www-mtl.mit.edu/~jimg/project_top.html)
The Low Energy Challenge

• In a cubic millimeter volume
  ⇒ current best battery: 1 Joule

• For continuous operation over 1 day:

  ⇓

  • power consumption ≃ 10 \( \mu W \)
  • efficient power management strategies needed
  • efficient networking algorithms required
RF vs Laser Communications

- **Radio Frequency:**
  - small mote size $\Rightarrow$ limited space for antennas $\Rightarrow$ extremely short-wavelength $\Rightarrow$ high-frequency transmissions
  - relatively complex circuits

- **Laser**
  - significantly lower power consumption (cheap, narrow beams emitted)
  - free line-of-sight paths required
  - passive (no power) optical transmission techniques (mirror-based retroreflection)
Sensors for a wide variety of conditions:

- temperature
- motion
- noise levels
- light conditions
- humidity
- object presence
- mechanical stress levels on attached objects
• continuous sensing
• detection of a crucial event
• location sensing (including motion tracking)
• local control of actuators (ambient intelligence)
• micro-sensing
Applications (III)

- Environmental applications
  - forest fire detection
  - flood detection (US Projects ALERT, COUGAR, DataSpace)
  - precision agriculture
  - weather forecast
  - planetary exploration

- Health applications
  - telemonitoring of human physiological data ("Health Smart Home", Grenoble, France)
  - tracking and monitoring doctors/patients
Applications (IV)

- Home applications
  - home automation (local/remote management of home devices)
  - smart environments (adapt to the people’s needs)
    (“Residential Laboratory”, Georgia Institute of Technology)

- More applications (partial list)
  - inventory control
  - security / military
  - vehicle tracking and detection
  - interactive museum (San Francisco Exploratorium)
  - monitoring material fatigue
Sensors vs Ad-hoc Wireless Networks

- much larger number of devices
- more dense deployment / interactions / complexity
- frequent failures
- more limited capabilities
- more frequently/more dynamically changing topology
- less global knowledge
New/Critical Challenges (I)

- **Scalability**
  - huge number of sensor nodes
  - high densities of nodes
  - many complex interactions

- Success rate: percentage of delivered data

  *How does protocol performance scale with size?*

  Even correctness may be affected by size.
• **Fault-tolerance**

Sensors may
- fail (temporarily or permanently)
- be blocked / removed
- cease communication
due to various reasons
- physical damage
- power exhaustion
- interference
- power saving mechanisms

Can the network *tolerate failures well?*
New/Critical Challenges (III)

- **Efficiency**
  - *energy* spent
  - *time* (for data propagation)
New/Critical Challenges (IV)

- Inherent *trade-offs* (e.g. *energy vs time*)
- *Competing goals* / various aspects:
  - minimizing total energy spent in the network
  - maximizing the number of “alive” sensors over time
  - combining energy efficiency and fault-tolerance
  - balancing the energy dissipation
- Application dependence
- *Dynamic* changes / heterogeneity
• variety of protocols needed / hybrid combinations
• adaptive protocols, locality
• simplicity, randomization, distributedness
Other Challenges

- production cost (goal: < $1 per node)
- hardware constraints
• Introduction (technology, applications, challenges, algorithmic performance properties)
• Network deployment, connectivity, coverage: models and metrics
• Data propagation algorithms I (data-centric, cluster-based)
• Data propagation algorithms II (greedy probabilistic forwarding, energy balance)
• Geographic routing and obstacle avoidance algorithms
• Localization and tracking
• Wireless characteristics (interference, energy cost models)
• Sleep-awake power saving and topology control schemes
• Medium access control (MAC) aspects and protocols
• Time synchronization
• Mobile sensor networks
• Implementation and engineering of algorithms for sensor networks
• Application development environments
A “canonical” problem: Local Event Detection and Data Propagation

A single sensor, \( p \), senses a local event \( E \). The general propagation problem is the following:

“How can sensor \( p \), via cooperation with the rest of the sensors in the network, propagate information about event \( E \) to the control center?”
Directed Diffusion (DD): a *tree-structure* protocol (suitable for low dynamics)

LEACH: *clustering* (suitable for small area networks)

Local Target Protocol (LTP): *local optimization* (best for dense networks)

Probabilistic Forwarding Protocol (PFR): *redundant optimized* transmissions (good efficiency / fault-tolerance trade-offs, best for sparse networks)

Energy Balanced Protocol (EBP): guaranteeing *same per sensor energy* (prolong network life-time)
Directed Diffusion

• requires *some coordination* between sensors
• creates / maintains some *global structure* (set of paths)
• a paradigm / suite of several protocols
• here: a *tree-based* version
Directed Diffusion elements

- *Interest* messages (issued by the control center)
- *Gradients* (towards the control center)
- *Data* messages (by the relevant sensors)
- *Reinforcements* of gradients (to select “best” paths)
An interest contains the description of a sensing task.
Task descriptions are named e.g. by a list of attribute-value pairs.
The description specifies an interest for data matching the attributes.

```
type = wheeled vehicle               // detect vehicle location
interval = 10 ms                     // send events every 10 ms
duration = 10 minutes                // for the next 10 minutes
rect = [-100, 100, 200, 400]         // from sensors within rectangle
```

Example of an interest

Interests are injected into the network at the control center
For each interest message received, a gradient is created.

Gradients are formed by *local interaction* of neighboring nodes, establishing a gradient towards each other.

Gradients store a *value* (data rate) and a *direction* (towards the sink) for “pulling down” data.
Data propagation

- A sensor that receives an interest it can serve, begins sensing.
- As soon as a matching event is detected, data messages are sent to the relevant neighbors using the gradients established.
- A data cache is maintained at each sensor recording recent history.
The sink initially repeatedly *diffuses an interest* for a low-rate event notification, through *exploratory* messages.

The *gradients created* by exploratory messages are called exploratory and have *low data rate*.

As soon as a matching event is detected, exploratory events are generated and *routed back to the sink*.

After the sink receives those exploratory events, it *reinforces one (or more) particular neighbor* in order to “draw down” real data.

The gradients that are set up for receiving high data rate information are called *data gradients*. 
To reinforce a neighbor, the sink re-sends the original interest message with a higher rate.

Upon reception of this message a node updates the corresponding gradient to match the requested data rate.

The selection of a neighbor for reinforcement is based on local criteria, i.e.:

- the neighbor that reported first a new event is reinforced.
- the higher data rate neighbor is reinforced.
- more than one neighbors are reinforced.

The data cache is used to determine which criteria are fulfilled.
Negative reinforcement is applied when *certain criteria* are met i.e. a gradient *does not deliver* any new messages for an amount of time, or a gradient *has a very low data rate*, etc.
Summary Evaluation of Directed Diffusion

- improves over flooding a lot
- significant energy savings
- performance drops when network dynamics are high
- The network is partitioned in clusters.
- Each cluster has one cluster-head.
- Each non cluster-head sends data to the head of the cluster it belongs to.
- Cluster-heads gather the sent data, compress it and send it to the sink directly.
Dynamic Clusters
• Node $n$ decides with probability $T(n)$ to elect itself cluster-head.
$P$: the desired percentage of cluster heads.
$r$: the current round.
$G$: the set of nodes that have not been cluster-heads in the last $\frac{1}{P}$ rounds.

• $T(n)$ is chosen so as to get on the average the same number of cluster-heads in each round.

$$T(n) = \begin{cases} 
\frac{P}{1 - P \cdot \text{rmod} \frac{1}{P}} & \text{if } n \in G \\
0 & \text{otherwise}
\end{cases}$$
Each cluster-head broadcasts an advertisement message to the rest nodes using a CSMA-MAC protocol.

Non cluster-head nodes hear the advertisements of all cluster-head nodes.

Each non cluster-head node decides its cluster-head by choosing the one that has the stronger signal.
• Each cluster-head is *informed for the members of its cluster*.

• The cluster-head creates a *TDMA schedule*

• The cluster-head *broadcasts the schedule back* to the cluster members.
Steady Phase

- Non cluster-heads
  - Sense the environment.
  - Send their data to the cluster head during their transmission time.

- Cluster-heads
  - Receive data from non cluster-head nodes.
  - Compress the data received.
  - Send their data directly to the base station.
- reduces energy dissipation through *compression of data at cluster-heads*.
- in large networks, *direct transmissions are very expensive*.
- performance drops when the network traffic is high (e.g. many events generated).
State of the art Surveys


T1. Challenges for Wireless Sensor Networks
T2. Models, Topology, Connectivity
T3. Localization, Time Synchronization, Coordination
T4. Data Propagation and Collection
T5. Energy Optimization
T6. Mobility Management
T7. Security Aspects
T8. Tools, Applications and Use Cases

- **DCOSS**: International Conference on Distributed Computing in Sensor Systems (http://www.dcoss.org/)
- **MASS**: International Conference on Mobile Ad-hoc and Sensor Systems
- **IPSN**: ACM/IEEE International Conference on Information Processing in Sensor Networks(http://ipsn.acm.org/)
- **SECON**: Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (*http://www.ieee-secon.org/*)

- Generic Theoretical Distributed Computing Conferences (*PODC, DISC, SPAA*, etc.)

- Generic Networks/Systems conferences (*MOBICOM, MOBIHOC, INFOCOM*, etc.)