Sensceive: Middleware for Wireless Sensor Networks

Waltenegus Dargie
Technical University of Dresden, Germany
AINA 2008, Okinawa, Japan
Senceive stands for Sense and Perceive
Outline

• Motivation

• Architecture

• Implementation

• Application

• Experience learned
Motivation

• Xu et al. and Chintalapudi et al. developed the Wisden platform for SHM.
  – The sensor network itself was established and tested with 25 sensor nodes on three floors of a medium-sized office building and on a seismic test structure for conducting experiments.

• Kim et al. deployed 64 Mica sensor nodes on the San Francisco Golden Gate bridge to study the reaction of the bridge to strong wind and earthquake.
Werner-Allan et al. propose wireless sensor networks for active volcano monitoring.

- They deploy a linear network of 16 sensor in central Ecuador to monitor seismic and infrasonic signals resulting from an active volcano.
- Each sensor node was equipped with a microphone and a seismometer. Interestingly, the sensor network could be able to capture 230 volcano events just over three weeks.
Motivation

- Stoianov et al. propose the PipeNet for monitoring large diameter, bulk-water transmission pipelines.
  - The network collects hydraulic and acoustic/vibration data at high-sampling rate.
Motivation

• The above networks are optimised according to the sensing tasks for which they are deployed.

• On the other hand, they exhibit significant similarities in
  – the types of sensor node hardware they use,
  – the frequency band and bandwidth of communication,
  – the runtime environment and
  – some aspects of sensing, processing, and communication as well.
Motivation

- Time synchronisation;
- High sampling rate for a short duration;
- High resolution of the sampled data;
- Multi-hop communication; and thereby the need for medium access and link control; and,
- Periodic sensing and periodic sleeping.
## Motivation

<table>
<thead>
<tr>
<th></th>
<th>Btnode 3</th>
<th>mica2</th>
<th>mica2dot</th>
<th>micaz</th>
<th>telos A</th>
<th>tmote sky</th>
<th>EYES</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Manufacturer</strong></td>
<td>Art of Technology</td>
<td>Crossbow</td>
<td></td>
<td></td>
<td>Imote iv</td>
<td></td>
<td>Univ. of Twente</td>
</tr>
<tr>
<td><strong>Microcontroller</strong></td>
<td></td>
<td>Atmel Atmega 128L</td>
<td></td>
<td></td>
<td>Texas Instruments MSP430</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Clock frequency</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.37 MHz</td>
<td>4 MHz</td>
<td>7.37 MHz</td>
<td>8 MHz</td>
<td>5 MHz</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>RAM (KB)</strong></td>
<td>64 + 180</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>10</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td><strong>ROM (KB)</strong></td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>60</td>
<td>48</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td><strong>Storage (KB)</strong></td>
<td>4</td>
<td>512</td>
<td>512</td>
<td>256</td>
<td>1024</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td><strong>Radio</strong></td>
<td>Chipcon CC1000 315/433/868/916 MHz 38.4 Kbauds</td>
<td>Chipcon CC2420 2.4 GHz 250Kbps IEEE 802.15.4</td>
<td>RFM TR1001868 MHz 57.6 Kbps</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Max Range (m)</strong></td>
<td>150-300</td>
<td></td>
<td></td>
<td>75-100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Power</strong></td>
<td>2 AA batteries</td>
<td>Coin cell</td>
<td></td>
<td></td>
<td>2 AA Batteries</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>PC connector</strong></td>
<td>Through PC-connected programming board</td>
<td>USB</td>
<td>Serial Port</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>OS</strong></td>
<td>Nut/OS</td>
<td>TinyOS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Transducers</strong></td>
<td>On acquisition board</td>
<td>On board</td>
<td>On acquisition board</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Extras</strong></td>
<td>+ Bluetooth radio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Motivation

• TinyOS
  – Lightweight program: 400 byte program storage and 50 byte RAM
  – Concurrency is supported
  – Layered components: communication between components based on command: down-call and events (up-calls)
Motivation

• Given the expense of sensor nodes and the density requirements as the development and test phase, present WSNs should
  – *Be shared* by multiple test applications;
  – Be flexible to be *reconfigured* and fine-tuned;
  and
  – Support *parallel testing* and experimentation
Architecture

Application

Data Gathering service

Configuration Service

Message service

Kernel

DB Manager
Architecture

- The Configuration Service provides the following functionalities:
  - Detailed network status information
  - Information about configuration aspects
  - Modify individual mote configuration
  - Modify global configuration
  - Manual mote adding and removal
• The Data Gathering Service provides the following functionalities:
  – Send snapshot queries
  – Send historical queries
  – Start long run queries with or without data listener
  – Provide information about available sensors in the network
  – Register listener for changes in network status
Implementation

- Initialise mote
- Set microphone gain
- Get data
- Delete data
- Start sleep mode
- Set sampling rate
- Get mote state details
Implementation

• Software constraints
  – TinyOS 2.x: no driver support to directly access each and every functionality of the sensor boards (MTS300 or MTS310)
  – The minimum sensing interval is about 200 ms, all sensors have to be sampled together and at the same time
  – Lack of resource arbitration, leading to unpredictable results if two data requests are processed at the same time.
Implementation

• A special focus was set in the hardware programming, on recording and analysing audio signal.
  – This includes for example calculation of noise level, zero crossing rate, signal energy, tone-detection and range measurement
Implementation
Application
Application

Meeting

Lecture

Party
Application
Room Activity

Reasoning

Feature Extraction
(for each sensor type)

Aggregation
(over each sensor type)

Area 1 of Sensors
Local processing

Area 2 of Sensors
Local processing

... Area n of Sensors
Local processing

Knowledge Base
(Context Ontology, Features, Rules, Belief Nets)
Implementation
Experience learned

• RF is highly unreliable: packet loss exceeds 60% in concrete walled buildings
• Almost each sensor node requires individual calibration (particularly accelerometer sensors)
• Sampling above 4Kbps renders DSP impossible (for audio)
• Dissemination of configuration parameters is very costly at present