Data Management in Sensor Networks

Ellen Munthe-Kaas
Jarle Søberg
Hans Vatne Hansen
Outline

Sensor networks
   Characteristics
   Motes
   Application domains
   Data management

TinyOS
TinyDB
Sensor Networks

Base station (gateway)

Motes (sensors)
Sensor Network Characteristics

Autonomous nodes
Small, low-cost, low-power, multifunctional
Sensing, data processing, and communicating components

Sensor network is composed of large number of sensor nodes
Proximity to physical phenomena
  • Deployed inside the phenomenon or very close to it

Monitoring and collecting physical data
No human interaction for weeks or months at a time
Long-term, low-power nature
Motes

Mica2 mote with 2 AA batteries (provide power for one year’s use)

Mica2DOT mote. Powered with button battery

Spec smart dust; total size 5 mm2

Mote: Short for remote. Refers to a wireless transceiver that is also a remote sensor.
Mote Hardware

Made up of four basic components

sensing unit
  - usually two subunits: sensor and ADC

processing unit
  - makes the sensor collaborate with the other nodes to carry out the assigned sensing tasks

transceiver unit
  - connects the node to the network

power unit
  - small, standard batteries
Motes for the mandatory assignment

**Mica2**

**Processor:** MPR400CB based on Atmel ATmega128L  
**Radio:** 900 MHz multi-channel transceiver  
**Memory:** 4 kB Configuration EEPROM  
128 kB Program Flash Memory  
512 kB Measurement (Serial) Flash  
**Power:** 2 x AA  
**OS:** TinyOS v1.0  
**Weight:** 18g (excluding batteries)  

**Sensors:**  
- Light  
- Temperature  
- Acoustic  
  (accelerometer)

**Actuators:**  
- Sounder
Motes vs. Traditional Computing

Embedded OS
  Usually an image flashed to the device

Lossy, ad hoc radio communication
  E.g. wrt slow CPU

Sensing hardware
Severe power constraints
Application Domains

Environmental
Health
Military
Commercial
Environmental Applications

Tracking the movements of birds, animals, insects
Monitoring environmental conditions that affect crops and livestock
Chemical/biological detection
Biological, earth, and environmental monitoring in marine, soil, and atmospheric contexts
Meteorological or geophysical research
Pollution study, precision agriculture, irrigation
Biocomplexity mapping of environment
Flood detection, forest fire detection
Health Applications

Integrated patient monitoring
   Body sensor networks

Telemonitoring of human physiological data
Tracking and monitoring doctors and patients inside a hospital
Tracking and monitoring patients and rescue personnel during rescue operations
Military Applications

Monitoring friendly forces, equipment and ammunition
Battlefield surveillance
Reconnaissance of opposing forces and terrain
Nuclear, biological and chemical (NBC) attack detection and reconnaissance
Commercial Applications

Monitoring product quality
Constructing smart office spaces
Interactive toys
Smart structures with sensor nodes embedded inside
Machine diagnostics
Interactive museums
Managing inventory control
Environmental control in office buildings
Detecting and monitoring car thefts
Vehicle tracking and detection
Tracking goods with RFID
Managing Data

**Purpose of sensor network:** Obtain real-world data
Extract and combine data from the network

**But:** Programming sensor networks is hard!
Months of lifetime required from small batteries
Lossy, low-bandwidth, short range communication
Highly distributed environment
Application development
Application deployment administration
Data Management Systems for Sensor Networks
Data Management System Challenges

Routing
Resource allocation
Deployment
Query language, query optimization
Outline

Sensor networks
TinyOS
TinyDB
TinyOS

Operating system for managing and accessing mote HW

Characteristics:
- Energy-efficient
- Programming model: Components
- Only one application running at a time
- No process isolation or scheduling
- No kernel
- No protection domains
- No memory manager
- No multithreading

Programming language: nesC
Outline

Sensor networks
TinyOS
TinyDB
TinyDB

High level abstraction
Data centric programming
Interact with sensor network as a whole
Extensible framework

Under the hood:
Intelligent query processing: query optimization, power efficient execution
Fault mitigation: automatically introduce redundancy, avoid problem areas

SELECT nodeid, light FROM sensors WHERE light > 400 SAMPLE PERIOD 1s
Query Language Essentials

Declarative queries
  Simple, SQL-like queries

  Users specify the data they want and the rate at which data should be refreshed

  Using predicates, not specific addresses

TinyDB collects data from motes in the environment, filters it, aggregates it, and routes it out to a PC. TinyDB does this with power-efficient in-network processing algorithms.
Data Model

Relational model
Single table sensors
  One column (attribute) per type of value that a device can produce (light, temperature,...)

One row (tuple) per node per instant in time

Physically partitioned across all nodes in the network

Tuples are materialized only at need and stored only for a short period or delivered directly to the network

Projections and transformations of tuples from sensors may be stored in materialization points
The sensors Table

sensors(epoch, nodeid, volume, light, temp, ...)

SELECT nodeid, light
FROM sensors
WHERE light > 400
SAMPLE PERIOD 1s
TinyDB Routing Tree

PC side

TinyDB GUI
TinyDB Client API
PostgreSQL DBMS

Mote side

0

TinyDB query processor

Sensor network

1 2 3 4 5 6 7
Communication Scheduling

A mote upon receiving a request to perform a query:
awakens

synchronizes its clock

chooses the sender of the msg as its parent

forwards the query, setting the delivery interval for children to be slightly before the time its parent expects to see the partial state tuple
Aggregates: Centralized Approach

Server-based approach: All sensor readings are sent to the base station, which then computes the aggregates

Example:

```sql
SELECT COUNT(*)
FROM sensors
```

How many transmissions?
Aggregates: Centralized Approach

When reviewing: Please note that this slide is only meaningful when animated. If you understand why, you have understood the concept.
Aggregates: Distributed Approach

In TinyDB aggregates are computed in-network whenever possible.
  Lower number of transmissions
  Lower latency
  Lower power consumption

Example:
  SELECT COUNT(*)
  FROM sensors

How may transmissions?
Aggregates: Distributed Approach

When reviewing: Please note that this slide is only meaningful when animated. If you understand why, you have understood the concept.


Literature

Samuel R. Madden, Michael J. Franklin, Joseph M. Hellerstein, Wei Hong:
*TinyDB: An Acquisitional Query Processing System for Sensor Networks*,

(Available through the ACM Digital Library; cf. http://x-port.uio.no)

Some additional reading:
   Samuel R. Madden, Michael J. Franklin, Joseph M. Hellerstein, Wei Hong:
   *TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks*,
   *OSDI*, 2002