15-744: Computer Networking

L-13 Sensor Networks

Sensor Networks
- Directed Diffusion
- Aggregation
- Assigned reading
  - TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks
  - Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks

Outline
- Sensor Networks
- Directed Diffusion
- TAG
- Synopsis Diffusion

Smart-Dust/Motes
- First introduced in late 90’s by groups at UCB/UCLA/USC
  - Published at Mobicom/SOSP conferences
- Small, resource limited devices
  - CPU, disk, power, bandwidth, etc.
- Simple scalar sensors – temperature, motion
- Single domain of deployment (e.g. farm, battlefield, etc.) for a targeted task (find the tanks)
- Ad-hoc wireless network
Smart-Dust/Motes

- Hardware
  - UCB motes
- Programming
  - TinyOS
- Query processing
  - TinyDB
- Directed diffusion
- Geographic hash tables
- Power management
  - MAC protocols
  - Adaptive topologies

Berkeley Motes

- Devices that incorporate communications, processing, sensors, and batteries into a small package
- Atmel microcontroller with sensors and a communication unit
  - RF transceiver, laser module, or a corner cube reflector
  - Temperature, light, humidity, pressure, 3 axis magnetometers, 3 axis accelerometers

Berkeley Motes (Levis & Culler, ASPLOS 02)

<table>
<thead>
<tr>
<th>Mote Type</th>
<th>nec</th>
<th>ren1</th>
<th>ren2</th>
<th>det</th>
<th>mini</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>9/09</td>
<td>10/00</td>
<td>6/01</td>
<td>8/04</td>
<td>2/02</td>
</tr>
<tr>
<td>Microcontroller</td>
<td>AT91SL555</td>
<td>ATmega128</td>
<td>ATmega128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prog. mem. (KB)</td>
<td>8</td>
<td>16</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAM (KB)</td>
<td>512</td>
<td>1</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nonvolatile storage</td>
<td>24LC256</td>
<td>256</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection type</td>
<td>EEp</td>
<td>55F</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq. (KB)</td>
<td>49</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Power source</td>
<td>L4</td>
<td>Aik</td>
<td>L1</td>
<td>Aik</td>
<td></td>
</tr>
<tr>
<td>Size</td>
<td>CR2450</td>
<td>2AA</td>
<td>CR2032</td>
<td>2AA</td>
<td></td>
</tr>
<tr>
<td>Capacity (mAh)</td>
<td>575</td>
<td>2850</td>
<td>225</td>
<td>2550</td>
<td></td>
</tr>
<tr>
<td>Communication</td>
<td>Radio</td>
<td>RPM, TTI1000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rate (Kbps)</td>
<td>19</td>
<td>29</td>
<td>39</td>
<td>49</td>
<td>49/49</td>
</tr>
<tr>
<td>Modulation type</td>
<td>OOK</td>
<td>10K</td>
<td>OOK/ASK</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sensor Net Sample Apps

- Habitat Monitoring: Storm petrels on great duck island, microclimates on James Reserve.
- Earthquake monitoring in shake-test sites.
- Vehicle detection: sensors along a road, collect data about passing vehicles.
- Traditional monitoring apparatus.
Metric: Communication

- Lifetime from one pair of AA batteries
  - 2-3 days at full power
  - 6 months at 2% duty cycle
- Communication dominates cost
  - < few mS to compute
  - 30mS to send message

Communication In Sensor Nets

- Radio communication has high link-level losses
  - typically about 20% @ 5m
- Ad-hoc neighbor discovery
- Tree-based routing

Outline

- Sensor Networks
  - Directed Diffusion
  - TAG
- Synopsis Diffusion

The long term goal

Embed numerous distributed devices to monitor and interact with physical world: in workspaces, hospitals, homes, vehicles, and “the environment” (water, soil, air…)

Network these devices so that they can coordinate to perform higher-level tasks.

Requires robust distributed systems of tens of thousands of devices.
Motivation

- Properties of Sensor Networks
  - Data centric, but not node centric
  - Have no notion of central authority
  - Are often resource constrained
- Nodes are tied to physical locations, but:
  - They may not know the topology
  - They may fail or move arbitrarily
- Problem: How can we get data from the sensors?

Directed Diffusion

- Data centric – nodes are unimportant
- Request driven:
  - Sinks place requests as interests
  - Sources are eventually found and satisfy interests
  - Intermediate nodes route data toward sinks
- Localized repair and reinforcement
- Multi-path delivery for multiple sources, sinks, and queries

Motivating Example

- Sensor nodes are monitoring a flat space for animals
- We are interested in receiving data for all 4-legged creatures seen in a rectangle
- We want to specify the data rate

Interest and Event Naming

- Query/interest:
  1. Type=four-legged animal
  2. Interval=20ms (event data rate)
  3. Duration=10 seconds (time to cache)
  4. Rect=[-100, 100, 200, 400]
- Reply:
  1. Type=four-legged animal
  2. Instance = elephant
  3. Location = [125, 220]
  4. Intensity = 0.6
  5. Confidence = 0.85
  6. Timestamp = 01:20:40
- Attribute-Value pairs, no advanced naming scheme
Diffusion (High Level)

• Sinks broadcast interest to neighbors
• Interests are cached by neighbors
• Gradients are set up pointing back to where interests came from at low data rate
• Once a sensor receives an interest, it routes measurements along gradients

Summary

• Data Centric
  • Sensors net is queried for specific data
  • Source of data is irrelevant
  • No sensor-specific query

• Application Specific
  • In-sensor processing to reduce data transmitted
  • In-sensor caching

• Localized Algorithms
  • Maintain minimum local connectivity – save energy
  • Achieve global objective through local coordination

• Its gains due to aggregation and duplicate suppression may make it more viable than ad-hoc routing in sensor networks

Outline

• Sensor Networks
  • Directed Diffusion
  • TAG
  • Synopsis Diffusion
TAG Introduction

- Programming sensor nets is hard!
- Declarative queries are easy
  - Tiny Aggregation (TAG): In-network processing via declarative queries
- In-network processing of aggregates
  - Common data analysis operation
  - Communication reducing
    - Operator dependent benefit
  - Across nodes during same epoch
- Exploit semantics improve efficiency!
- Example:
  - Vehicle tracking application: 2 weeks for 2 students
  - Vehicle tracking query: took 2 minutes to write, worked just as well!

Basic Aggregation

- In each epoch:
  - Each node samples local sensors once
  - Generates partial state record (PSR)
    - local readings
    - readings from children
  - Outputs PSR during its comm. slot.
- At end of epoch, PSR for whole network output at root
- (In paper: pipelining, grouping)

Illustration: Aggregation

- SELECT COUNT(*) FROM sensors
- Slot 1
- Slot 2

Illustration: Aggregation

- SELECT MAX(mag) FROM sensors WHERE mag > thresh
  - EPOCH DURATION 64ms
Types of Aggregates

- SQL supports MIN, MAX, SUM, COUNT, AVERAGE

- Any function can be computed via TAG

- In network benefit for many operations
  - E.g. Standard deviation, top/bottom N, spatial union/intersection, histograms, etc.
  - Compactness of PSR
Taxonomy of Aggregates

- **TAG insight:** classify aggregates according to various functional properties
  - Yields a general set of optimizations that can automatically be applied

<table>
<thead>
<tr>
<th>Property</th>
<th>Examples</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partial State</td>
<td>MEDIAN: unbounded, MAX: 1 record</td>
<td>Effectiveness of TAG</td>
</tr>
<tr>
<td>Duplicate Sensitivity</td>
<td>MIN: dup. insensitive, AVG: dup. sensitive</td>
<td>Routing Redundancy</td>
</tr>
<tr>
<td>Exemplary vs. Summary</td>
<td>MAX: exemplary, COUNT: summary</td>
<td>Applicability of Sampling, Effect of Loss</td>
</tr>
<tr>
<td>Monotonic</td>
<td>COUNT: monotonic, AVG: non-monotonic</td>
<td>Hypothesis Testing, Snooping</td>
</tr>
</tbody>
</table>

Optimization: Channel Sharing (“Snooping”)

- **Insight:** Shared channel enables optimizations

  - **Suppress messages that won’t affect aggregate**
    - E.g., MAX
    - Applies to all exemplary, monotonic aggregates

Benefit of In-Network Processing

**Simulation Results**
- 2500 Nodes
- 50x50 Grid
- Depth = ~10
- Neighbors = ~20

**Total Bytes Xmitted vs. Aggregation Function**

- Some aggregates require dramatically more state!

Optimization: Hypothesis Testing

- **Insight:** Guess from root can be used for suppression
  - E.g. ‘MIN < 50’
  - Works for monotonic & exemplary aggregates
    - Also summary, if imprecision allowed

- **How is hypothesis computed?**
  - Blind or statistically informed guess
  - Observation over network subset
Optimization: Use Multiple Parents

- For duplicate insensitive aggregates
- Or aggregates that can be expressed as a linear combination of parts
  - Send (part of) aggregate to all parents
    - In just one message, via broadcast
  - Decreases variance

Multiple Parents Results

- Better than previous analysis expected!
- Losses aren't independent!
- Insight: spreads data over many links

Outline

- Sensor Networks
- Directed Diffusion
- TAG
- Synopsis Diffusion

Aggregation in Wireless Sensors

Aggregate data is often more important
In-network aggregation over tree with unreliable communication

Used by current systems:
- TinyDB [Madden et al. OSDI’02]
- Cougar [Bonnet et al. MDM’01]

Not robust against node- or link-failures
Traditional Approach

- Reliable communication
  - E.g., RMST over Directed Diffusion [Stann’03]
- High resource overhead
  - 3x more energy consumption
  - 3x more latency
  - 25% less channel capacity
- Not suitable for resource constrained sensors

Exploiting Broadcast Medium

- Robust multi-path
- Energy-efficient
- Double-counting
- Different ordering

Challenge: order and duplicate insensitivity (ODI)

A Naïve ODI Algorithm

- Goal: count the live sensors in the network

Synopsis Diffusion (SenSys’04)

- Goal: count the live sensors in the network

Approximate COUNT algorithm: logarithmic size bit vector

Synopsis should be small

Challenge
**Synopsis Diffusion over Rings**

- A node is in ring $i$ if it is $i$ hops away from the base-station.
- Broadcasts by nodes in ring $i$ are received by neighbors in ring $i-1$.
- Each node transmits once = optimal energy cost (same as Tree).

**Evaluation**

Approximate COUNT with Synopsis Diffusion

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Energy (mJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree</td>
<td>41.8</td>
</tr>
<tr>
<td>Syn. Diff.</td>
<td>42.1</td>
</tr>
</tbody>
</table>

- More robust than Tree
- Almost as energy efficient as Tree

Per node energy