ENGINEERING SENSOR NETWORKS FOR LONG-TERM ENVIRONMENTAL MONITORING

Guest lecturer: Doug Carlson
Hi. Who are you?

• I’m Doug Carlson, one of Dr. Terzis’s PhD students.
• I have been doing research on wireless sensing systems and communication protocols at JHU here since 2008.
• You guys are going to be discussing my paper on Thursday.
What are you doing here?

• Dr. Chang is out of town this week.
What are we doing here?

Learning stuff?

More specifically, answering the question:

How can we build efficient and scalable systems for long-term environmental monitoring (LTEM)?
The Plan

• Discuss the LTEM application area and some usage scenarios we have encountered.
• Talk about the inherent challenges of this application.
• Discuss potential HW/networking approaches to tackle them.
• Present the solutions that me and my collaborators have developed in this area.
BACKGROUND

What are we working with?

- Limited hardware budget, favors mote-class devices
- Battery powered
- Radio range of 10’s to 100’s of meters
  - Tends to be largest contributor to power budget
- Local data buffering in flash
- Some fixed number of ADC inputs
LTEM Application Requirements

- Sample periodically (~O(minutes))
- Collect data wirelessly
- Loose latency requirements (hours or days)
- High yield requirements (>99%)
- Survive on batteries for months or years

Background

(Actually clusters of 4-32 analog sensors)
What challenges do you foresee?

(Actually clusters of 2-8 sensors)
Hardware Challenges

1. Necessary radio communication ranges vary widely across the network.
2. Some deployments favor uniform sensor coverage, some favor dense clusters.
3. In a large deployment, how do you keep track of what goes where?
Networking Challenges

1. Coordinating a large, patchy network.
2. Low power disconnected operation.
3. Link state collection and routing.
HARDWARE

HW problem 1: Variable Communication

Range Requirements

1. What options do we have?
2. What are the costs/benefits of adding a second communication channel (wifi, cell modem, etc)?
3. How might each of these options impact our energy budget?

(Actually clusters of 4-32 analog sensors)
Our Solution: modular RF front-end

- **Leaf** configuration
  - +10 dBm output power
  - -90 dBm RX threshold
  - ~3 dB gain with PCB antenna

- **Router** configuration
  - PA/LNA, external antenna,
  - +26 dBm output
  - -98 dBm RX threshold
  - Free-space range of 700m – 24km
Problem 2: Sensor Allocation

• MCU has a fixed number of ADC inputs.
• How can you effectively support both sparse and dense sensor needs?

20 Telos motes, most within ~5 m groups

Node       Relay       Uplink

Diagram showing 20 Telos motes分布在~5 m groups.
Our solution: digital, chainable MUX board

- Connects up to 8 analog sensors to each slave MCU.
- I2C digital communication
- Automatic device discovery
HW Problem 3: Sensor metadata and provenance

• How do you interpret raw voltage measurements?
• How do you keep track of sensor calibrations?
• How do you associate data streams with physical locations?
Our Solution: Store types/IDs on MCU’s

- Barcode **ALL THE THINGS**
- Reflect this information in flash.
- Send this information back with sensor samples
- Android app to scan barcodes and record their physical locations
Networking

Networking Problem 1: Coordinating large, patchy networks.

• Given the hardware we’ve discussed, how might you structure your network? Your collection protocol?
• How does the addition of a new node affect the duty cycle of the rest of the network?
Larger Networks: More Data Forwarding
Larger Networks: More Data Forwarding
Larger Networks: More Data Forwarding
Our solution: separate router tier from leaf tier

- Leaves buffer data in flash.
- Leaves push outstanding data to router when woken up.
- Routers buffer leaf data in flash.
- Routers push outstanding data to base station when woken up.

Separate channel for each patch.
Separate channel for routers.
Networking Problem 2: Disconnected Operation

- What should leaf nodes do when their router disappears?
- What should routers do when the base station disappears?
- How can you re-associate nodes to the network?
Low Power Probing

• You guys talked about this, right?

Why don’t 1 and 2 collide?
How can we get the most out of this?

- What factors dictate the selection of a probe interval?
- How do probe interval and wakeup frequency affect duty cycle?
- When should a node send probes? When shouldn’t it?
- How would you adapt LPP to support multiple channels?
Networking Problem 3: Link state collection and routing

- What information do you need to figure out where to send data?
- How does the idle radio behavior impact collecting this data?
- What factors may lead to a mismatch between link quality and route reliability?
- What impact does bad routing information have on the network?
Impact of link quality assessment on the bottom line

**Cub Hill Duty Cycle and Contacts**

Estimated DC: 1.67%

Hard-to-capture effects present

Link quality freshness matters

Figure 1
Impact of instability on route selection

![Bar chart showing failed connections for different days and network periods. The chart compares failed connections for "Healthy" and "Rebooting" motes over the period from 10/28/2009 to 11/30/2009.](chart.png)
How can we deal with poor link quality assessment?

- “Plan B is to make Plan A work”
- Moar data
- …?

What is the **minimum** amount of link quality information that we can get by with?
Do we even need link quality information?

• How would you get data from a leaf to a router without knowing local LQ or distances?
• What are the issues associated with this?
• Can we do better, perhaps by using some… trickery?
Networking approach: Non-destructive Concurrent Transmissions

• Recall the LPP example!
• This worked because hardware-generated acks are very deterministic and interfere non-destructively.
Symbol Boundaries
How can we do the same with data packets?

• Careful timing
  • We have access to a 26 MHz crystal
  • We can use a timer capture module to record the time that the packet preamble is sent or received

• Modify packet contents deterministically
  • OK to increment hop-count, decrement TTL, compute checksums, and apply forward error correction.
Solution 1: Try ALL paths at the same time

Solution 1: Try ALL paths at the same time

Solution 1: Try ALL paths at the same time

Solution 1: Try ALL paths at the same time

Solution 1: Try ALL paths at the same time

Solution 1: Try ALL paths at the same time

Can we do better than flooding?

- NO!
- jk yeah
Which of these nodes are actually useful?

- B
- E
- A
- F
- D
- C
- G
Solution 2: Identify Useful Forwarders with CXFS

w is on some shortest path from S to D if
\[ d(S,w) + d(w,D) = d(S,D) \]

- **Burst Setup**
  - S Floods
  - D measures \( d(S,D) \)
  - w measures \( d(S,w) \)

- **Setup**
  - D Floods
  - D tells \( d(S,D) \) to w, S.
  - w measures \( d(D,w) \)
  - w assumes \( d(w,D) \)

- **Acknowledgement**
  - D Floods
  - D tells \( d(S,D) \) to w, S.
  - w measures \( d(D,w) \)
  - w assumes \( d(w,D) \)

- **Data Forwarding**
  - S Floods
  - w forwards if \( d(S,W) + d(w,D) \leq d(S,D) \)
  - OTHERWISE
  - w sleeps.
Distance Metrics: "You go to war with the army you have"

Must Fit Multi-TX Context

Obeys triangle inequality (on a given flood)

Low-State (one number per node)

Mostly-Symmetric

Why not use hop count?

What does RSSI mean? How do you get local information?

Augment with "boundary zone," optionally smooth it.

Thursday’s paper has details and evaluation.
Our Testbed

Red: Forwards more frequently

Close Source
Our Testbed

Red: Forwards more frequently

Distant Source
Implementation Details

Each node gets a "slot" for their data.

Each slot is broken into “frames” where data may be sent.
What else is left?

- Somewhere for the data to go
- Recovery for lost packets
- Tools to configure nodes
- Tools for end users (setup, download, monitoring)
Questions/comments?