Applications: Part I

Jun Yang
CPS 296.1, Spring 2007
Sensor Data Processing

Announcements (Jan. 16)

- A note on lecture notes
  - No need to copy anything that appears on slides
- Reading and participation
  - 10%: mini-reviews of papers on reading list
    - Announced via email and posted on Web
    - One review due next Tuesday
  - 20%: presentation of two papers in class
    - 30 minutes each; need to prepare slides, and meet me in advance
    - Sign-up for the first round will begin next week
- Participation is important!
  - For those of you who are auditing, I encourage you to present and participate in discussion nevertheless

Three applications

- Two environmental monitoring applications
  - Great Duck Island
  - Redwoods in Sonoma, California
- One urban application
  - CarTel: distributed, mobile sensors on cars

App 1: Great Duck Island

  - You should also check out their follow-up evaluation two years later, in Robert Szewczyk, Alan Mainwaring, Joseph Polastre, John Anderson, David Culler. "An Analysis of a Large Scale Habitat Monitoring Application." International Conference on Embedded Networked Sensor Systems, 2004

Scientific goals

- Field research on Leach’s Storm Petrel on Great Duck Island
  - What is the usage pattern of nesting burrows?
  - How do environmental parameters change over time inside burrows and in surrounding areas?
  - What are the differences in the micro-environments with and without large numbers of nesting burrows?
    - I.e., what factors make for a good nest?

Why sensors?

- Less expensive
  - Imaging the cost involved for humans to take measurements at different sites every 5-10 minutes or every hour, over 9 months!
- Least intrusive
  - Human presence disturbs the environment being measured
  - These birds will likely desert their nesting burrows if disturbed by humans during the first 2 weeks of incubation
System architecture

Armed with more power, solar panels, and better antennas, responsible for forwarding collected data to base station

Laptop at ranger’s station with satellite connection to Internet

KISS

- Simple collection task: sample the environment at each node and report readings to base station, at a pre-determined rate
  - Can apply data compression to reduce transmission cost
- Simple communication network
  - Started with two-level, single-hop communication
    - Between each node and gateway
    - Between each gateway and base station
  - Later moved to multiple-hop communication
    - Lower power requirement, larger patches

Power constraint

Power is not replenishable on sensor nodes
Gateways need solar panels, but large ones are intrusive
  - Low duty cycle required to make the system last
    - 9 month = working at most 1.4 hours per day = 5.8% duty cycle
    - Wake up, sample, communicate, and sleep for a long time
    - Schedule communication carefully
      - Nodes listen at the right time (based on “levels”) for forwarding to work
  - Use low-power states as much as possible
    - Deep sleep better than sleep better than idle
  - Sensing uses power too
    - Some require long startup time before reading stabilizes
      - E.g.: 500ms for temperature vs. up to 30000ms for humidity

Data logging

- Everything transmitted/forwarded is logged
  - Sensor nodes have some local Flash storage
  - Base station records everything in a database
  - Gateways’ capabilities fall in between
- Why?
  - Communication failures are common
    - In addition to random interference, there may be dead nodes downstream, or thunderstorms
  - Logging at an intermediate node allows it to take “custody” of data and retransmit on the source’s behalf
  - Logging allows data to be recovered eventually

Re-tasking

- In situ with gizmos or over the network
- Forms of re-tasking
  - Parameter (e.g., sampling rate) adjustments
  - Virtual-machine program updates
  - Binary code image updates
- Cost increases with flexibility
- How about benefit?

Example of data collected

- What else can you conclude from this data?
- Can you do better than periodically reporting the temperature?
- Data has redundancy, and application-specific utility
App 2: California Redwoods


Scientific goals

- Study the microclimate over the volume of an entire redwood tree (~70m)
  - Substantial variation over height
  - Interesting dynamics over time

Why sensors?

- State of the art: haul heavy instruments up the tree, and collect data through very long serial cable connected to a battery-powered data logger
  - Costly and difficult to do over longer time or at fine spatial resolution
- With sensors
  - High-resolution data from 33 nodes every 5 minutes over 44 days ("a month in the life of a redwood tree"), close to 1 million data points
  - Low maintenance overhead

Deployment

- Temperature, humidity, PAR (photosynthetically active radiation), etc.
- Sample and report raw readings every 5 min. (awake 4 sec. = 1.3% duty cycle)
- Multi-hop network to a PC-class gateway
  - Powered by batteries, solar panel
  - Cellular link to Internet
- Data logging everywhere (again!)

Calibration and planning

Lots of work before actual deployment

- Determine what to measure
  - E.g., tested total solar radiation and barometric sensors, and decided they were not as useful
- Plan deployment locations, duty cycle, etc.
- Calibrate on roof and in weather chamber

Unexpected things still occurred

- E.g., log filled up before the target end date, because data from calibration tests was not purged—not discovered until the nodes were collected from the tree

Outlier detection

- Anomalous readings are common
- High correlation with battery failure
  - Automatic outlier rejection based on operating battery voltage range
- Some manual cleaning is still required
  - How can we make this process easier?
Data yield

- Data yield over the network < from local logs
  - Latter is not affected by communication failure
  - Logging does help (and could have done better if logs did not fill up)

High-dimensional data

- What else can you conclude from this data?
- Can you do better than periodically reporting readings?
- Data has redundancy over both time and space

PAR readings

- Random over a day? How about two?

Improvement since GDI

- Hardware, system software, and infrastructure have all become much better
- More focus on good “science,” e.g., calibration, data analysis
- We still need better control over the system
  - System monitoring/re-tasking is critical
  - Why didn’t they clear the logs after calibration?
  - More hooks for application to drive optimization
    - Signal-specific compression
    - Application-specific information utility

App 3: CarTel: mobile sensors on cars


Use cases

- Road traffic analysis:
  - Analyze commute time
  - Track traffic hot spots in real-time
  - Acquire street-level images
- Wide-area WiFi
  - “Borrow” people’s wireless services as you drive by the neighborhood!
- Automotive diagnostics
  - Analyze driving patterns for energy efficiency, emission levels, etc.
  - Collect on-board diagnostic data in real-time, e.g., engine load, air intake temperature, etc.
What is different

- Behold the power of power
  - With a moving car, you can power a Linux box with 1GB of flash memory, WiFi, GPS, PostgreSQL!
- Behold the power of presence—it’s a good thing to be in a civilized world
  - Existing infrastructure: cars, WiFi access points, GPS
- Behold the power of community
  - Lots of traffic, WiFi-enabled households, camera phones, and gadget lovers willing to mule your data around

A different challenge

- Continuous queries over intermittently connected data sources via a delay-tolerant network
  - Not always connected
  - Connection may not last long
- Node queries continuously generate data to be delivered
- Portal queries run over whatever data that has been successfully delivered so far
  - Make effective use of every direct or indirect (via mules) connection opportunity
    - FIFO result delivery is suboptimal
    - Allow portal/nodes to prioritize delivery based on app. needs

Prioritization examples

- Local prioritization
  - Node has been collecting its GPS locations over time
  - When connecting with portal, node prioritizes transmission of a subset of time/location data that would allow portal to interpolate its trajectory
- Global prioritization
  - Node has been collecting its speed averages on roads A, B, …
  - Portal needs the latest overall speed information about all roads
    - Say it has fewer or less recent reports about B than A
  - When node connects with portal, portal tells node to prioritize transmission of data about B (over A)

Visions of future sensors diverging?

- “Smart dust” (e.g., for environmental monitoring) vs. sensors networked with more traditional approaches (e.g., for pervasive computing)
- Different target applications and operation conditions result in wildly different systems
  - Is “sensor networks” even a coherent field?

What about similarities?

- What are the resource constraints?
- Is prioritization in CarTel so different from figuring what to transmit and what to log in environmental monitoring?
  - Notion of application-specific information utility?
- How about controlling local data acquisition/transmission to formulate an answer to a global question?
- Fusion of sensor data across time, space, modalities?
- Exploiting redundancy/correlation in data?
- Accepting approximation and uncertainty?
  - Many fundamentals remain common