TASK in Redwood Trees

Wei Hong (IRB)
Sam Madden (IRB, MIT)
Rob Szewczyk (UCB)
Jan. 14, 2003
Acknowledgement

- TASK Team
  - Alan Broad (Xbow)
  - Phil. Buonadonna (IRB)
  - Anind Dey (IRB)
  - David Culler (UCB, IRB)
  - David Gay (IRB)
  - Joe Hellerstein (IRB, UCB)
  - Alan Mainwaring (IRB)
  - Jaidev Prabhu (Xbow)
  - Joe Polastre (UCB)
Outline

- TASK Overview
- Status Report
- Progress in Making TASK Real
- TASK/Deployment Roadmap
- The Redwood Tree Deployment
- Data From the Redwood Trees
- Calibration Results
- Conclusions
TASK Overview

- TASK = Tiny Application Sensor Kit, formerly GSK, ASK.
- Sensor Network in a Box: rapid sensor network deployment for *non-computer scientists*
- Tools suite built on top of TinyDB
  - Sensor metadata management
  - Query configuration
  - Network monitoring
  - Data visualization
  - Integration with DBMS and data analysis tools
TASK Architecture

External Tools

JDBC/ODBC

DBMS (PostgreSQL)

TASK Client Tools

JDBC

Internet

Sensornet Appliance

TASK Server

TinyDB Sensor Network

TASK Field Tools
Status Report

- Released with TinyOS 1.1!
  - Install the task-tinydb package
  - apps/TASKApp, tools/java/net/tinyos/task
  - http://berkeley.intel-research.net/task

- Successful deployments in Lab and UCBG redwood trees
  - Largest deployment: ~80 weather station nodes
  - Network longevity: 4-5 months
Progress in Making TASK Real

- Power Management
- Time Synchronization
- Improved Query Sharing
- Watchdog
- Improved Routing Layer
Power Management

Coarse-grained app-controlled communication scheduling

Epoch (10s - 100s of seconds)

Mote ID

1 2 3 4 5

... zzz ... ... zzz ...

2-4s Waking Period
Power Management (cont)

- **Benefits**
  - Can still use CSMA within waking period
    - No reservation required: new nodes can join easily!
  - Minimal code changes from previous code base without power management

- **Drawbacks**
  - Longer waking time vs. TDMA?
    - Could stagger slots based on tree-depth
  - No “guaranteed” slot reservation
    - Nothing is guaranteed anyway

- **Improvement**
  - Adaptively setting waking period (currently hardwired to 4s)
    - Network size
    - Sensor startup + acquisition time
Time Synchronization

- All messages include a 5 byte time stamp indicating system time in ms
  - Synchronize (e.g. set system time to timestamp) with
    - Any message from parent
    - Any new query message (even if not from parent)
  - Punt on multiple queries
  - Timestamps written just after preamble is xmitted
- All nodes agree that the waking period begins when (system time % epoch dur = 0)
  - And lasts for WAKING_PERIOD ms
- Adjustment of clock happens by changing duration of sleep cycle, not wake cycle.
- If node hasn’t heard from it’s parent for $k$ epochs
  - Switch to “always on” mode for 1 epoch
Improved Query Sharing

- Viral propagation of query messages (each query many messages)
  - Compensate for loss of query messages
  - New nodes joining the network

- Previous implementation
  - Each node asks for all query messages whenever a query result with unknown query id is snooped
  - Jams network!

- Improved implementation
  - Query request message contains bitmap of messages already received
  - Shut up if a neighbor has just requested for the same query
Stopping a query

- Must stop query on all nodes at the same time, or query rekindles
- Solution:
  - Explicitly notify neighbors of “dead” queries
  - Don’t share “dead” queries
Watchdog

- New watchdog component
- Timer set to multiples of epoch duration
- Watchdog reset every time a data message is heard during an epoch
- Watchdog triggers when no data messages are heard in multiple epochs.
- Key: motes always resetable remotely!
TASK/Deployment Roadmap

- Distributions of Do-It-Yourself kits, Q1’04
  - Stargate-based gateway appliance
  - VB/HTML-based improved GUI tools
  - PDA field tool
  - New generation packaging of motes
- Support for high data rate applications, Q2’04
  - Take over Intel fab vibration monitoring application
  - GGB app?
- Evolve into core software infrastructure for all Intel Research pilot projects
  - HP data center
  - SAP asset tracking
  - Etc.
- Port to iMote
The Redwood Tree Deployment

- Collaboration with Prof. Todd Dawson
- Collect dense sensor readings to monitor climatic variations across
  - altitudes,
  - angles,
  - time,
  - forest locations, etc.
- Versus sporadic monitoring points with 30lb loggers!
- Current focus: study how dense sensor data affect predictions of conventional tree-growth models
Data from the Redwood Trees

Relative humidity at different heights

- Relative humidity (%)
- Date
Redwood trees data

Temperature at different heights

Date

Temperature difference (°C)

Temperature (°C)
Towards Gradient analysis

Average temperature difference between top and bottom of the tree
Temperature

Average temperature difference between 30 feet and bottom of the tree

Temperature difference (°C)

Time of day
Relative humidity

Average RH difference between top and bottom of the tree

Time of day

Relative humidity difference (%)
Relative humidity difference between 30 feet and bottom of the tree
Data from the Redwood Trees

**Relative humidity at different heights**

- **Date**: 08/04, 08/05, 08/06, 08/07, 08/08, 08/09, 08/10

**Relative humidity (%)**

- **08/04**: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 0

**Humidity Difference (%)**

- **08/04**: -30, -20, -10, 0, 10, 20, 30

Legend:
- Blue: 10
- Green: 20
- Red: 30
- Cyan: 34
- Magenta: 40
The Calibration Process

- Growth chamber calibration of temperature, humidity, light against trusted sensor
- VLSB rooftop calibration of PAR sensor against trusted sensor
Redwood trees data

Temperature at different heights

Date

Temperature (°C)

Temperature difference (°C)

Date

10
20
30
34
40
Towards Gradient analysis

Average temperature difference between top and bottom of the tree
Temperature

Average temperature difference between 30 feet and bottom of the tree
Relative humidity

Average RH difference between top and bottom of the tree

Time of day

Relative humidity difference (%)
Relative humidity

Average RH difference between 30 feet and bottom of the tree

Relative humidity difference (%)
Chamber data -- reference

Ground truth data, growth chamber, Dec. 1, 2003

- PAR (µmol m⁻² s⁻¹)
- Temperature (°C)
- RH (%)
Mote 69 data, growth chamber, Dec. 1, 2003

Time (s)

PAR (µmol m⁻² s⁻¹)

Temperature (°C)

RH (%)
Temperature measurements

Measured Temperature v. Reference Temperature, Chamber Dec. 2003

- Mote 69
- Mote 65
- Ideal fit
Temperature error distribution

CDF of temperature errors

Fraction of samples

Temperature difference (°C)

December 2003 chamber run
Bias of 1°C
Relative humidity measurements

![Graph showing measured vs. reference humidity for Motes 69 and 65. The data points form a scatter plot with a trend line, indicating a correlation between the two readings.](image)
Relative humidity measurements

December 2003 chamber run
Errors computed assuming a constant sensor offset of 13.25%
PAR Measurements – regression

Calibrated Data
Fit Data
Difference

fit = f(par, v, 1/v, par/v)

PAR (µmol m^(-2)s^(-1)) -200 0 200 400 600 800 1000 1200 1400 1600
Time
18:00 00:00 06:00 12:00 18:00 00:00 06:00 12:00 18:00
PAR Measurements

Measured vs. calibrated PAR

fit = f(par, v, 1/v, par/v)

Calibrated vs. Our Data
Ideal fit
PAR measurements

CDF of PAR errors

fit = f(par, v, 1/v, par/v)

Error magnitude (µmol m\(^{-2}\)s\(^{-1}\))

Fraction of readings
Still a ways to go...

Converted PAR data

Date

08/04 08/05 08/06 08/07 08/08 08/09

Converted PAR (µmol m⁻² s⁻¹)

0 500 1000 1500 2000
Conclusions

- TASK readily available in TinyOS1.1
- Proven to work well in low-data-rate, environmental monitoring type of applications
- Love to get more “customers”
- Need more developers
- Have lots of data, love to share