Infrastructures for Power Profiling

Jan Beutel, ETH Zurich
Sensornets Are Hard

- Sensor networks often fail/operate poorly
  - Great Duck Island network: median yield 58% [SenSys 2004]
  - Redwood network: median yield 40% [SenSys 2005]
  - Volcano network: median yield 68% [OSDI 2006]

- Survey of causes
  - Protocol conflicts/interference
  - Collisions and congestion induced loss
  - Neighbor management (with layer 2 scheduling, e.g. TMAC)
  - Don’t know!
  - Low-power, limited resources make complete logging prohibitively expensive...
Infrastructures for Power Profiling

The PermaSense Application
PermaSense Project – Alpine Permafrost Monitoring

- Cooperation with Uni Basel (C. Tschudin) and Uni Zurich (S. Gruber)
PermaSense Site – The Matterhorn

- Located in Zermatt, CH
- 4478 m
- Site of recent rockfall due to extreme warming (07/2003)

  - 25 nodes
  - Different sensors
    - Temperature, rock electrodes, rock motion, ice and water pressure
  - −40 to +65°C
  - Rockfall, snow and ice, avalanches
  - 30 min. duty-cycle
  - 3 years unattended lifetime
PermaSense Site – The Deployment Region

Hörnli Ridge
3500 m asl
PermaSense Scientific Goals – Validation of Models

© Stephan Gruber
[Uni Zurich, Physical Geography]
PermaSense – System Architecture

- Shockfish TinyNode
  - TI MSP430 – 48K Flash
  - Xemics 868 MHz Radio
- Custom DAQ Board
- Single Battery (Li-SOCl2)
  - 300 uA average power budget
- Ruggedized
PermaSense – Complex Sensor Installations
Infrastructures for Power Profiling

A Methodology Driven Approach
Methodology and Development Tools

Extending the Logical View
- Detailed physical monitoring
- Control of the environment
- Physical stimulation
- Control of resources

From Platform to Testbed to Multi-Platform
- Native execution
- Log file analysis
- Influence of the environment

Advanced Software Engineering Practices
Automated Building and Testing

- Run Build and Test Loop Continuously
  - Watch repository for changes
  - Run build and tests
  - Publish status of the build and tests to developers

- Benefits
  - Detect errors early
  - Monitor code quality over time
  - Prevent integration problems
Regular Builds with Statistics

Number of Build Attempts: 296
Number of Broken Builds: 74
Number of Successful Builds: 222

Timeline of build types

Breakdown of build types

Timeline of coding violations
Profiling – Graphical Reporting of Results

TinyOS Flash Usage

TinyOS Ram Usage
Testbed – The Deployment-Support Network

Testbed Functions

- Remote reprogramming
- Extraction of log data
- Analysis
- Regression testing
Today's WSN Design and Development

**Simulation**
- TOSSIM [Levis2003]
- PowerTOSSIM [Shnayder2004]
- Avrora [Titzer2005]

**Virtualization and Emulation**
- EmStar [Ganesan2004]
- BEE [Chang2003, Kuusilinna2003]

**Test Grids**
- moteLab [Werner-Allen2005]
- Emstar arrays [Cerpa03/04]
- Kansei [Dutta2005]

Regular and Detailed Power Traces?

Figure abridged from D. Estrin/J. Elson
Getting Physical – Power Profiling Testbed Extension

Compilations ➔ Testing ➔ Reporting

- Mean Current Consumption
  - Graph showing current consumption over time with annotations:
    1. Applied to reduce energy consumption
    2. Applied to increase robustness

- Mean Current Consumption
  - Graph showing current consumption over time with different devices:

NCCR MICS National Competence Center in Research Mobile Information and Communication Systems
Detailed Power Traces and Analysis
Today's WSN Design and Development

**Simulation**
- TOSSIM [Levis2003]
- PowerTOSSIM [Shnayder2004]
- Avrora [Titzer2005]

**Virtualization and Emulation**
- EmStar [Ganesan2004]
- BEE [Chang2003, Kuusilinna2003]

**Test Grids**
- moteLab [Werner-Allen2005]
- Emstar arrays [Cerpa03/04]
- Kansei [Dutta2005]

Can we Emulate Reality in the Lab?

Figure abridged from D. Estrin/J. Elson
Physical Characterization Requirements

• Emulating the Environment ...
  – Temperature Cycle Testing (TCT)

• ... and Different Resource Utilization
  – Power sources: Batteries, rechargeable cells, photovoltaic cells, fixed DC power, noisy sources
  – Different (dis-)charging profiles
  – Depleted cells
Physical Characterization Architecture

[Diagram showing physical characterization architecture with components such as power supply with analyzer, signal or clock generator, switch unit, and devices like Tmote, TinyNode, MicaZ, DSN.]
Power Profiling – Trends and Detailed View

- Continuous Integration
- Testbed Infrastructure
- Physical Characterization
Automating Profiling and Validation

• Assertions Based on Reference Traces/Specification

\[
f_i(t) = \begin{cases} 
a_0 + a_1 \cdot x + \ldots & \text{if } t \in [t_{i-1}, t_i) \\
0 & \text{if } t \notin [t_{i-1}, t_i)
\end{cases}
\]

\[
f^-_i(t) = \begin{cases} 
f_i(t) - \Delta y^- & \text{if } t \in [t_{i-1}, t_i) \\
0 & \text{if } t \notin [t_{i-1}, t_i)
\end{cases}
\]

\[
f^-(t) = \sum_{i=1}^{n} f^-_i(t)
\]

\[
\forall i \in [-\Delta t, \Delta t], \forall i \in \mathbb{N}:
\]

\[
f^-(t + \tau(t_k)) = \begin{cases} 
f^-(t^-) & \text{if } -f(t^-) + f(t^+) \leq 0 \\
f^-(t^+) & \text{if } -f(t^-) + f(t^+) > 0
\end{cases}
\]

The upper bound \( f^+ \) follows accordingly with a bound value \( \Delta y^+ \).

• Integrated with each Build (Regression Testing)
Simultaneous Profiling on 2 Nodes
Detailed Analysis Reveals Dangerous Voltage Drops

\[ \text{Vcc}_{\text{min}} = 2.8 \text{ V} \]
4 Nodes – Zooming On The Details

- TCT Temp
- Room Temp
- Current Node 1
- Current Node 2

Continuous Integration
Physical Characterization
Testbed Infrastructure

Graphs showing power consumption for different nodes and temperature readings.
The Good, The Bad and the Ugly...

- The testing strategy seems to work.
- Need for multi-dimensional testing is apparent.
- Some (mis-) behavior seems to be systematic, and worse, it could not be observed on all nodes!
Remember the time when “compilers” didn’t yet have error/warning messages?
A Vision of Future WSN Design Methodology...

- (Live) Backannotation into the Design Space
  - "Closed loop" system design
  - Including live data from
    - Testbeds
    - Deployments

- Allowing to Refine and Check Architectures, Models and Implementation...
Thank You.