Title
SYS 5: Systems Infrastructure

Authors
Kevin Chang
John Hicks
Martin Lukac
et al.

Publication Date
2006

Center for Embedded Networked Sensing

Systems Infrastructure

Kevin Chang, John Hicks, Martin Lukac, Dustin McIntire, Tom Schoellhammer, Thanos Stathopoulos, Karen Weeks, Richard Guy

CENS Systems Laboratory

Problem space: Assembling complete deployments from variety of available components

The Systems Infrastructure team assembles, tests, and provides complete sensor network solutions containing both exploratory and hardened components, from high-level applications and analysis tools, down to hardware at the sensor platform level. We assist domain science teams with deployment planning and execution.

Solution highlights: Six selected tools used routinely in CENS deployments and testbeds

ESS: Extensible Sensing System

ESS Components

- **TinyOS:** Provides a scheduling system and the underlying CC1000 radio stack
- **MDA300 Driver:** Provides an interface to sample various sensors attached to an MDA300 sensor board
- **Routing layer:** An interchangeable layer to transport data packets to a central micro server
- **Time synchronization:** Works through the routing layer to provide reliable mote time stamping
- **DTN:** A persistent data buffer that works above the transport layer to provide in-network data storage and retransmission
- **Sympathy:** Provides system status information and fault isolation
- **Data Sampling Engine:** Allows a user to remotely program motes with queries to periodically return sensor data

Areas of Focus

- **Worst-case conne|ctivity requirements**
  - Science-driven placement of nodes
- **Continuous interactivity with motes**
  - Especially during installation
- **Energy versus robustness**
- **Vertical integration**
  - Sensor to microserver as well as microserver to database
- **Real-time visibility**
  - To adjust individual sensor placement and alignment

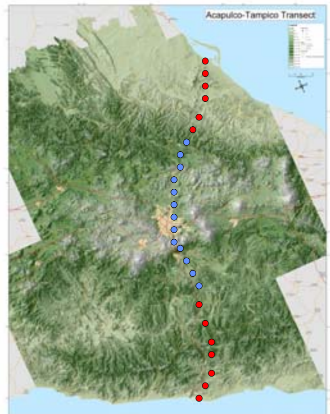
DTS: Disruption Tolerant Shell

Routine system management tasks in deployed WSNs

- Modifying data acquisition configuration
- System status information, disk space, and file counts
- Data file integrity, deploying new software
- Changing scheduling of system tasks

Disruption Tolerant Shell (DTS)

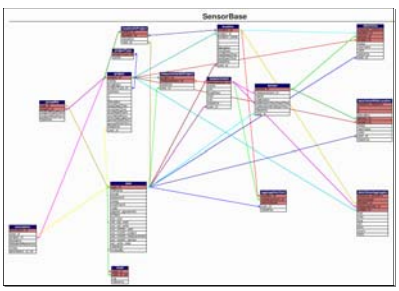
- Asynchronous remote shell interface to all nodes simultaneously
- No routing required
- No end to end connections required
- Ensures exactly one execution of a series of commands on all nodes
- Provides centralized collection of responses (can view responses from any node)
- Ensures that commands will succeed: as long as there is eventually a connection between a node and any other node which already has a command
- Uses a reliable and efficient publish-subscribe mechanism to disseminate shell commands and responses "epidemically" and reliably hop by hop
- Provides status client feature
- Provides file distribution feature



SensorBase.org

Data management challenges in deployed WSNs

- Different sensor networks use different data push mechanisms:
  - Different file formats
  - Different repository/servers (username/pw)
- Difficult to cull, parse, interpret from different sources
- Difficult to "google" for data sets
- Difficult to share/publish/annotate data sets



Sensorbase.org goals

A repository for sensor network data

- Easy to publish data
- Easy to share (permission/control) data
- Easy to search for data
- Flexible data formats

Search engine-like queries

- Natural language-like queries:
  - "get all the data points from user '%Richard%'"
  - "get all the data points from project 'Cold Air Drainage' from 2006-01-01 to 2006-01-05"
  - "get x,y,rawValue from project 'Botanical Garden'"
- The front-end also allows applications to use the REST/stateless HTTP GET requests to retrieve data points
  - Outputs text [and soon, xml] for users to interface with GnuPlot, MS Excel, DAS, etc.
  - interface with applications such as Google Maps, Google Earth, and even other web services and data management systems.

EmStar

The Problem

- Mote-class WSNs are challenging
  - Inherent computation/communication constraints
  - Heterogeneity (integrating motes with microservers)
- Combination yields networks that are unusually hard to design, develop, debug, deploy, maintain

The Solution

- A framework that allows a simple development path from simulation to deployment of mote-class WSNs
- Fault isolation through multiple processes
  - EmRun management tool provides robustness
- Modular design
- Visualization and debug tools
  - EmSim
  - EmView
  - EmTOS

Future directions

- Additional tool and module development
- Interface improvements
- Documentation and usability
- Port to other platforms

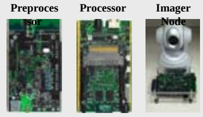
Layer 4: Extra Tools Help run, maintain, and debug application
Layer 3: Existing Modules, Services Existing useful components for applications
Layer 2: Device Patterns &Libraries IPC mechanism for a variety of interactions
Layer 1: Glib Handle events on IPC
Layer 0: FUSD Low Level IPC

DAS: Deployment Analysis System

LEAP: Low power, Energy-Aware Platform

LEAP Systems

EMAP Preprocessor — Processor — LEAP Imager Node

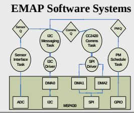


Adaptive Power States

EMAP Power State	Supply Current at 5V
Suspend (LPM4)	385µA
Current sensing task and idle task (LPM3)	419µA
Previous with all power domains enabled	862µA
Previous with 1.8V RF power enabled	950µA
Previous with all power domains and supplies enabled	7.6mA

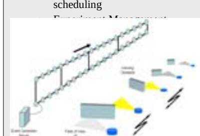
LEAP PXA Processor Power State	Supply Current at 5V
EMAP Only	0.4 - 1 mA
Suspend	5mA
Operating System Idle Task	41ma
Operating System Idle with SD Card Based Filesystem	50mA
Operating System Idle with Ethernet without traffic	48mA
iperf execution with Ethernet at maximum throughput	140mA
Execution at 100 percent computing load	165mA

EMAP Software Systems



Tools and Testbed

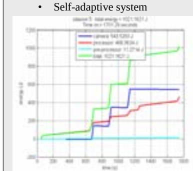
- **LEAP Node Systems**
  - Processor-Preprocessor coordination
  - Energy management
  - Sensor Interfaces
- **LEAP Testbed**
  - Physical Event Generator
  - User access control and scheduling



- **LEAP Emulator System**
  - Complete processor, preprocessor, energy, communication and sensing system verification

Results and Applications

- **Energy Aware Event Detection**
  - Micropower event detection sensors
  - Imager event identification
  - Distributed solution
  - Self-adaptive system



- Environmental Monitoring
- Actuated Sensing Platform
- Energy Aware Microserver
- Biomedical Monitoring