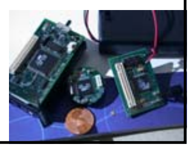
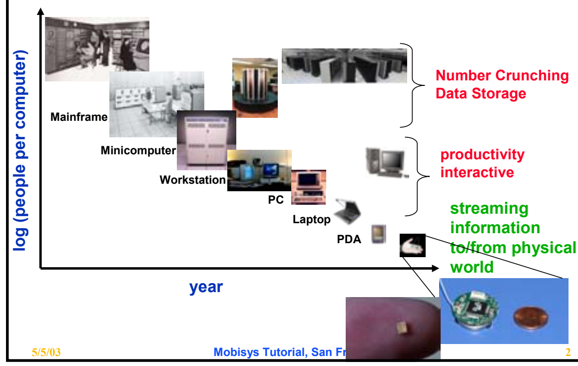
# Building Sensor Networks with TinyOS


**David Culler, Phil Levis, Rob Szewczyk, Joe Polastre**
*University of California, Berkeley*
**Intel Research Berkeley**

**http://webs.cs.berkeley.edu**

## New Class of Computing



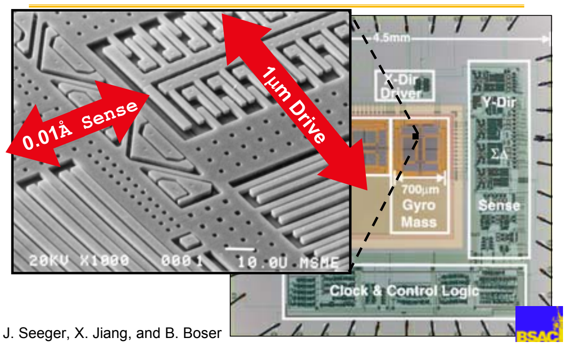



## Technology Push

- **CMOS miniaturization**
  - 1 M trans/$ => tiny (~mm$^2$), inexpensive processing and storage
  - 1-10 mW active, 1 $\mu$W passive (at 1% use 100 $\mu$W ave)
- **Micro-sensors (MEMS, Materials, Circuits)**
  - acceleration, vibration, gyroscope, tilt, magnetic, heat, motion, pressure, temp, light, moisture, humidity, barometric
  - chemical (CO, CO2, radon), biological, microradar, ...
  - actuators too (mirrors, motors, smart surfaces, micro-robots)
- **Communication**
  - short range, low bit-rate, CMOS radios (1-10 mW)
- **Power**
  - batteries remain primary storage (1,000 mW*s/mm$^3$), fuel cells 10x
  - solar (10 mW/cm$^2$, 0.1 mW indoors), vibration (~uW/gm), flow
- **1 cm$^3$ battery => 1 year at 10 msgs/sec**



## MEMS Gyroscope Chip

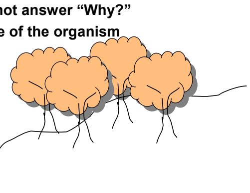



J. Seeger, X. Jiang, and B. Boser



## Application Pull

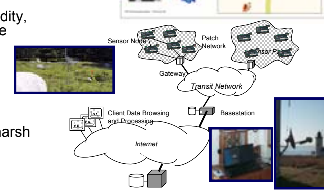
- **Env. Monitoring, Conservation biology, ecophysiology, ...**
  - discrete clustering of growth patterns
  - continuous changes in underlying environmental conditions
  - interpolation from a sensors cannot answer "Why?"
  - ⇒need instrumentation on the scale of the organism
  - ⇒minimize monitoring perturbance





## Example: Environment Monitoring

- Canonical "patch" net architecture
- live & historical readings www.greatduckisland.net
- 43 nodes, 7/13-11/18
- above and below ground
- light, temperature, relative humidity, and occupancy data, at 1 minute resolution
- >1 million measurements
  - Best nodes ~90,000
- 3 major maintenance events
- node design and packaging in harsh environment
  - -20 – 100 degrees, rain, wind
- power mgmt and interplay with sensors and environment

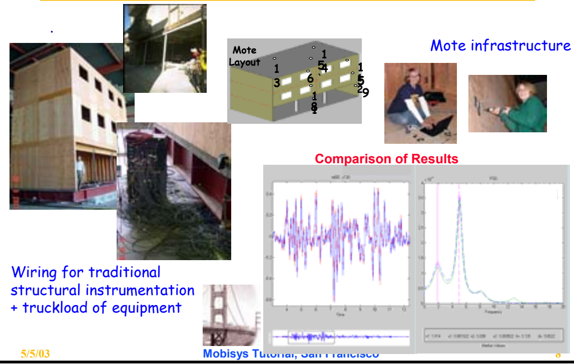

Mainwaring, Polastre, Szewczyk

## Application Pull

- Env. Monitoring, Conservation biology, ...
  - *Precision agriculture,*
  - *built environment comfort & efficiency ...*
  - *alarms, security, surveillance, treaty verification ...*
- Civil Engineering: structures response
  - instrumentation dominated by cost of “wires”
  - dense instrumentation to resolve interactions between components
  - continuous monitoring of response to detect fatigue
  - interactive, remote inspection after event



## Structural performance due to multi-directional ground motions (Glaser)



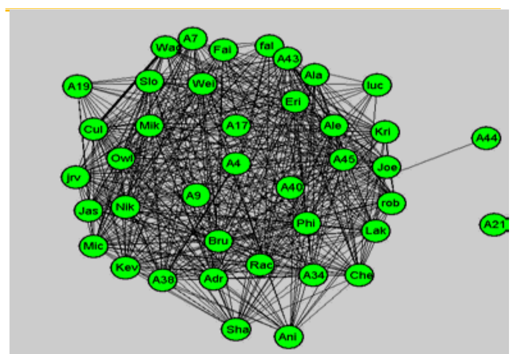



## Application Pull


- Monitoring Environments
  - habitat monitoring, conservation biology, ...
  - *Precision agriculture, land conservation, ...*
  - *built environment comfort & efficiency ...*
  - *alarms, security, surveillance, treaty verification ...*
- Monitoring Structures and Things
  - *structural response, condition-based maintenance*
  - *disaster management*
  - *urban terrain mapping & monitoring*
- Interactive Environments
  - manufacturing, asset tracking, fleet & franchise
  - context aware computing, non-verbal communication
  - assistance
    - » *home/elder care*
- Integrated robotics



## Meeting Social Network






## Distributed Control

Hill, Whitehouse, et al.

- Sensor field quietly monitors
  - net env. data for health monitoring
  - off-line geographic localization
- Event detected locally by small group
- Local broadcast of processed data
- Group leader elected to aggregate
  - time sensitive
- Multihop geographic transport to well-define dest
  - base-station for processing & response
  - mobile pursuer node
- Additional time-sensitive cooperative localization data for pursuer navigation and planning

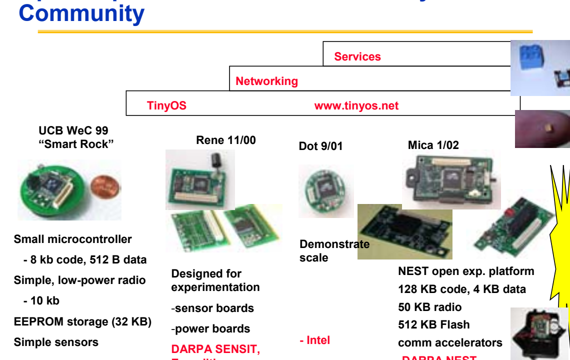


## Open Experimental Platform to Catalyze a Community

Services

Networking

TinyOS www.tinyos.net



UCB WeC 99 “Smart Rock”

Small microcontroller
- 8 kb code, 512 B data

Simple, low-power radio
- 10 kb

EEPROM storage (32 KB)

Simple sensors

Rene 11/00

Designed for experimentation

-sensor boards

-power boards

DARPA SENSIT, Expeditions

Crossbow

Dot 9/01

Demonstrate scale

- Intel

Mica 1/02

NEST open exp. platform

128 KB code, 4 KB data

50 KB radio

512 KB Flash

comm accelerators

-DARPA NEST

-200 research groups

## TinyOS/MICA Platform Users (ca 6/02)

- **ACCENTURE**
- **ALLEN, ANTHONY**
- **ALTARUM**
- **BAE SYSTEMS CONTROLS**
- **BALBOA INSTRUMENTS**
- **CARNEGIE MELLON UNIV**
- **CENTRID**
- **CLEVELAND STATE UNIV**
- **CORNELL UNIVERSITY**
- **DARTMOUTH COLLEGE**
- **DOBLE ENGINEERING COMPANY**
- **DUKE UNIVERSITY**
- **FRANCE TELECOM R&D**
- **GE KAYE INSTRUMENTS, INC**
- **GEORGE WASHINGTON UNIV.**
- **GEORGIA TECH RESEARCH INT**
- **GE**
- **GRAVITON, INC**
- **HONEYWELL**
- **HRL ABORATORIES**
- INTEL CORPORATION
- INTEL RESEARCH
- JPL
- KENT STATE UNIVERSITY
- LAWRENCE BERKELEY NAT'L
- LLNL
- LOS ALAMOS NATIONAL LAB
- MARYLAND PROCUREMENT
- MIT
- MITRE CORP.
- MSE TECH. APPLICATION INC
- NASA LANGLEY RESEARCH CTR
- NAT'L INST OF STD & TECH
- NICK OLIVAS LOS ALAMOS NA
- NORTH DAKOTA STATE UNIV
- PENNSYLVANIA STATE UNIV
- PHILLIPS
- ROBERT BOSCH CORP.
- RUIZ-SANDOVAL, M.E.
- RUTGERS STATE UNIVERSITY
- SANDIA NATIONAL LABS
- SIEMENS BUILDING TECH INC
- SILICON SENSING SYSTEMS
- SOUTHWEST RESEARCH
- TEMPLE UNIVERSITY
- **UNIV SOUTHERN CALIFORNIA**
- **UNIVERSITY OF CALIFORNIA**
- **UNIVERSITY OF CINCINNATI**
- **UNIVERSITY OF COLORADO**
- **UNIVERSITY OF ILLINOIS**
- **UNIVERSITY OF IOWA**
- **UNIVERSITY OF KANSAS**
- **UNIVERSITY OF MICHIGAN**
- **UNIVERSITY OF NOTRE DAME**
- **UNIVERSITY OF SOUTHERN CA**
- **UNIVERSITY OF TEXAS**
- **UNIVERSITY OF UTAH**
- **UNIVERSITY OF VIRGINIA**
- **US ARMY CECOM**
- **USC INFORMATION SCIENCES**
- **VANDERBILT UNIVERSITY**
- **VIGILANZ SYSTEMS**
- **VITRONICS INC**
- **WASHINGTON UNIVERSITY**
- **WAYNE STATE UNIVERSITY**
- **WILLOW TECHNOLOGIES LTD**
- **WJM, INC**
- **XEROX**
- **CENS @ UCLA**



## Typical Characteristics

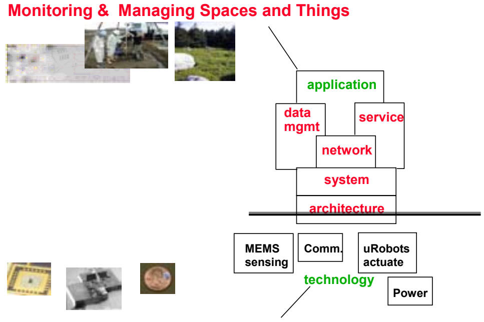
- # nodes >> # people
- sensor/actuator data stream
- unattended, inaccessible
- prolonged deployment
- energy constrained
- operate in aggregate
- in-network processing is necessary
- what they do changes over time

⇒must be self-organized, self-maintaining and programmed *in situ* to operate at very low duty cycle.



## The Challenges

Monitoring & Managing Spaces and Things



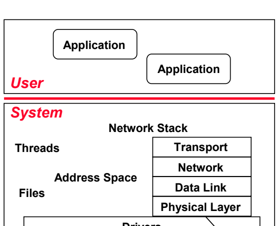

Miniature, low-power connections to the physical world



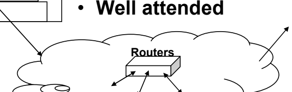
## An Operating System for Tiny Devices embedded in the Physical World



## Traditional Systems




- Well established layers of abstractions
- Strict boundaries
- Ample resources
- Independent Applications at endpoints communicate pt-pt through routers
- Well attended






## by comparison ...

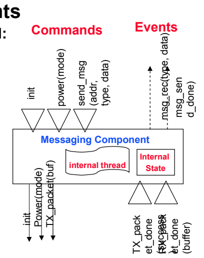
- Highly Constrained resources
  - processing, storage, bandwidth, power
- Applications spread over many small nodes
  - self-organizing Collectives
  - highly integrated with changing environment and network
  - communication is fundamental
- Concurrency intensive in bursts
  - streams of sensor data and network traffic
- Robust
  - inaccessible, critical operation
- Unclear where the boundaries belong

=> Provide a framework for:
- Resource-constrained concurrency
- Defining boundaries
- Appl'n-specific processing

allow abstractions to emerge

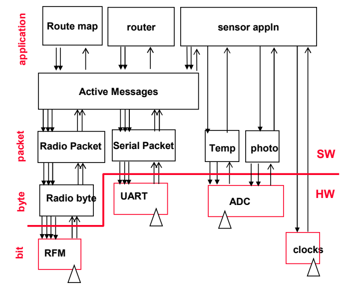
# Tiny OS Concepts

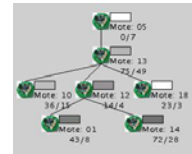
- Scheduler + Graph of Components
  - constrained two-level scheduling model: threads + events
- Component:
  - Commands,
  - Event Handlers
  - Frame (storage)
  - Tasks (concurrency)
- Constrained Storage Model
  - frame per component, shared stack, no heap
- Very lean multithreading
- Efficient Layering






# Application = Graph of Components



Example: ad hoc, multi-hop routing of photo sensor readings

3450 B code
226 B data

Graph of cooperating state machines on shared stack



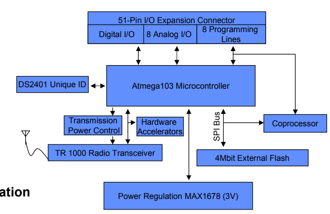
# Example TinyOS study

- UAV drops 10 nodes along road,
  - hot-water pipe insulation for package
- Nodes self-configure into linear network
- Synchronize (to 1/32 s)
- Calibrate magnetometers
- Each detects passing vehicle
- Share filtered sensor data with 5 neighbors
- Each calculates estimated direction & velocity
- Share results
- As plane passes by,
  - joins network
  - upload as much of missing dataset as possible from each node when in range
- 7.5 KB of code!
- While servicing the radio in SW every 50 us!



# The MICA architecture

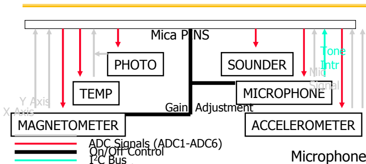
- Atmel ATMEGA103
  - 4 Mhz 8-bit CPU
  - 128KB Instruction Memory
  - 4KB RAM
- 4 Mbit flash (AT45DB041B)
  - SPI interface
  - 1-4 uj/bit r/w
- RFM TR1000 radio
  - 50 kb/s – ASK
  - Focused hardware acceleration
- Network programming
- Rich Expansion connector
  - i2c, SPI, GIO, 1-wire
  - Analog compare + interrupts
- TinyOS tool chain
- sub microsecond RF synchronization primitive

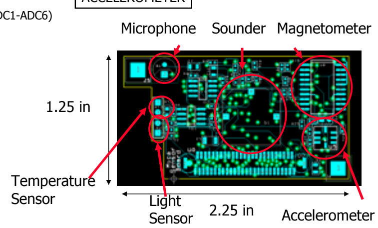



2xAA form factor



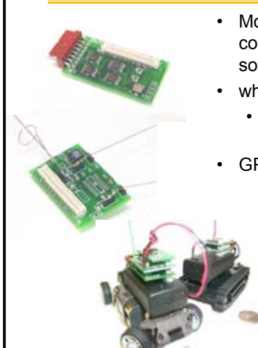
# Rich Sensor board






# Actuators



- Motor-Servo board interfaces any combination of two motors, servos, and solenoids to a toy car platform
- whisker board for obstacle detection
  - digital accelerometer (ADXL202) board for crude odometry
- GPS Board

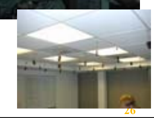
## Other Platform Components

- Crossbow
  - Mica 2 with ChipCon Radio
  - Mica Dot in small form factor
- Dust Inc
  - Blue – TI MSP & ChipCon
- Intel
  - Mica in PCM form factor
  - xscale + mica gateway
- many have spun their own from designs on the web
  - http://webs.cs.berkeley.edu/tos/hardware/hardware.html
- Sensor Boards
  - Basic protoboard
  - weatherstation (light, PAR, temp, humidity, barometer)
  - Ultrasound
  - Magnetometer
  - PIR
  - AC charger



## Networking

- Hands-on Experience with Large Networks of Tiny Network sensors
  ⇒intense constraints, freedom of abstraction
- Re-explore entire range of networking issues
  - encoding, framing, error handling
  - media access control, transmission rate control
  - discovery, multihop routing
  - broadcast, multicast, aggregation
  - active network capsule (reprogramming)
  - localization, time synchronization
  - security, network-wide protection
  - density independent wake-up and proximity est.
- Fundamentally new aspects in each





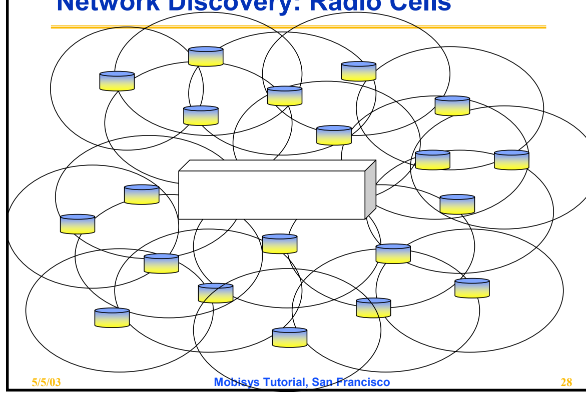
## The Nodes are the Infrastucture

- Simple Epidemic Algorithm Schema
  if (new mcast) then
  take local action
  retransmit modified request
- Examples: Network wakeup, command propagation
  - Build spanning tree
    - » record parent
- Naturally adapts to available connectivity
- Minimal state and protocol overhead

=> surprising complexity in this simple mechanism

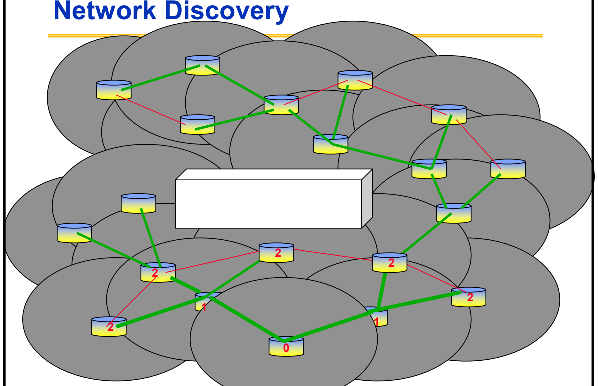


## Network Discovery: Radio Cells



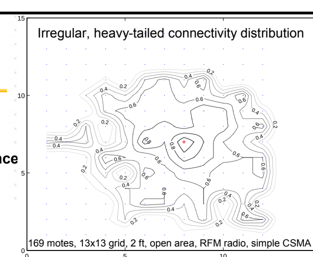


## Network Discovery



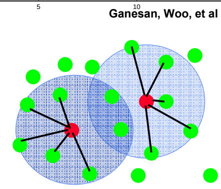


## Challenges




Ganesan, Woo, et al

- Uncertainty
  - Probabilistic Connectivity
  - Contention creates interference
  - Mobility
  - changing environment
- Power & Delay
  - Radios mostly OFF
- Limited bandwidth
- Discovery
  - Flooding does not scale
- Mechanism for reinforcement and suppression
- Naming, operators



## Programming Challenges

- thousands of constrained nodes,
- interacting in real-time with physical world,
- where you cannot touch them,
- and what you want them to do changes with time...

- How do you program the network?
- How do you specify what you want it to do?



## Programmable network fabric

- Architectural approach
  - new code image pushed through the network as packets
  - assembled and verified in local flash
  - second watch-dog processor reprograms main controller
- Viral code approach
  - each node runs a tiny virtual machine interpreter
  - captures the high-level behavior of application domain as individual instructions
  - packets are "capsule" sequence of high-level instructions
  - capsules can forward capsules
- Rich challenges
  - security
  - energy trade-offs
  - DOS

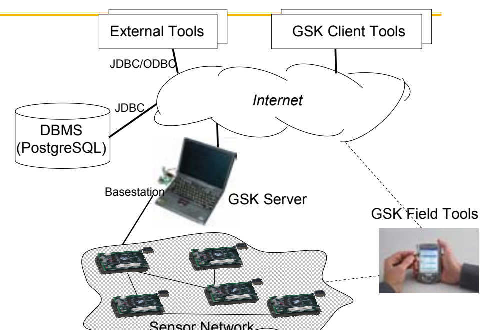


## Higher-level Programming?

- Ideally, would specify the desired global behavior
- Compilers would translate this into local operations

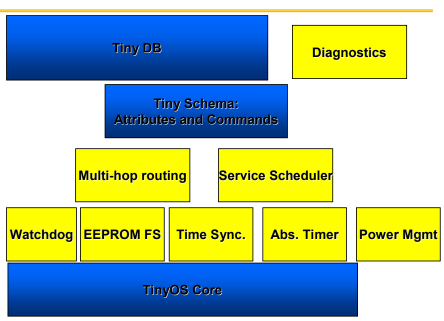


## GSK Architecture






## GSK Mote Side Components






## TinyDB Query Processor

- Multiple Simultaneous Declarative Queries:
  - Sensor Query: SELECT nodeid, humidity, temp FROM sensors SAMPLE PERIOD 8000
  - Health Query: SELECT nodeid, parent, depth, voltage FROM sensors SAMPLE PERIOD 16000
- In-network filtering and aggregation
- Routing Layer Abstraction: multihop and bcast (multiple implementations available)
- Only wake up for a fraction of each sample period
- TimeSync keeps sample periods aligned across nodes

## GSK Server

- Provide Remote (Java) API to client tools
- Inject queries and commands into sensor network
- Result dispatch from sensor network and to interested clients
- Log all data, queries and commands to DBMS
- Metadata management
  - Mote locations and maps
  - Mote capabilities: type of sensors
  - Mote characteristics: power consumption
  - Sensor calibration/conversion parameters



## A rich & growing research agenda

- Principles of Self-Organization
- Extremely Low-power topology mgmt & Routing
- Distributed Algorithms with Probabilistic Information
- Localization
- High-fidelity Time-Synchronization
- Distributed Control
- Auto-Calibration
- Collaborative Signal Processing
- Interpreting Spatially Distributed Data
- Multi-Resolution storage
- Simulation & Evaluation Methodologies
- Security
- Privacy



## Companies in the area

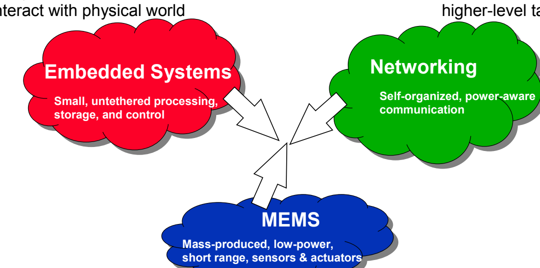
- Crossbow
- Dust Inc.
- Digital Sun
- ConnecTerra
- Millennial
- Graviton
- Sensoria
- Ember
- RF Monolithics, ChipCon, Atmel, MicroChip, ...
- Motorola, Honeywell, Bosch, Accenture, Philips...



## Confluence of Technologies

Many devices monitor and interact with physical world

Coordinate and perform higher-level tasks




Exploit spatially and temporally dense coupling to physical world



## Small Technology, Broad Agenda

- Social factors
  - security, privacy, information sharing
- Applications
  - long lived, self-maintaining, dense instrumentation of previously unobservable phenomena
  - interacting with a computational environment
- Programming the Ensemble
  - describe global behavior, synthesis local rules that have correct, predictable global behavior
- Distributed services
  - localization, time synchronization, resilient aggregation
- Networking
  - self-organizing multihop, resilient, energy efficient routing
  - despite limited storage and tremendous noise
- Operating system
  - extensive resource-constrained concurrency, modularity
  - framework for defining boundaries
- Architecture
  - rich interfaces and simple primitives allowing cross-layer optimization
- Components
  - low-power processor, ADC, radio, communication, encryption, sensors, batteries