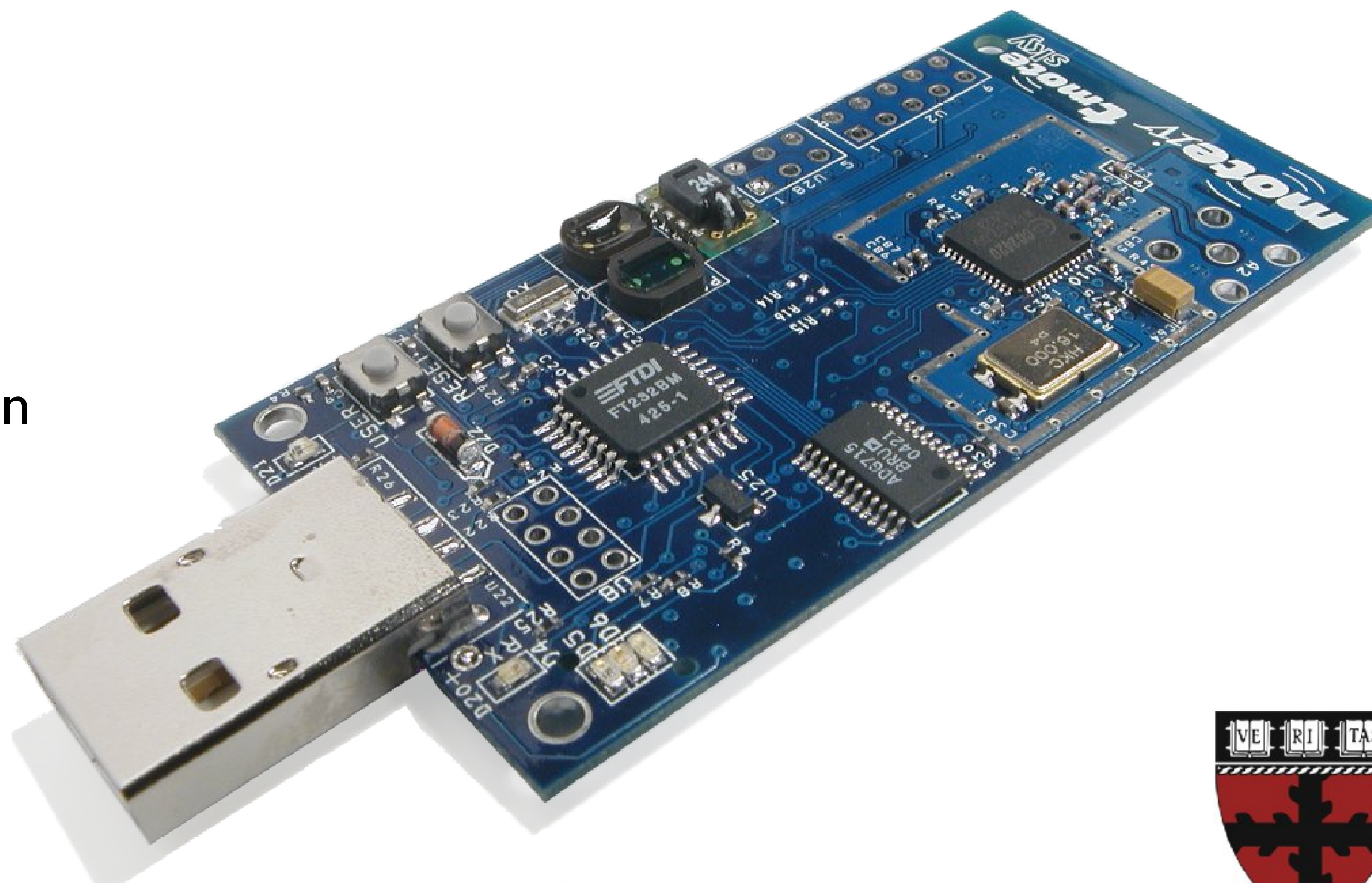
CS263

Wireless Sensor Networks

Lecture 1: Introduction

Prof. Matt Welsh
Harvard University

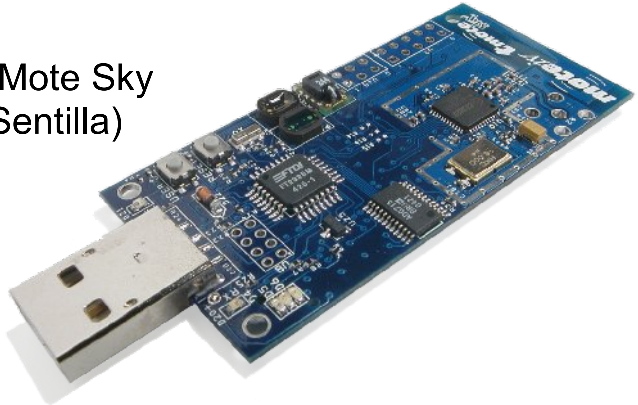
January 29, 2009



Introduction: Wireless Sensor Networks

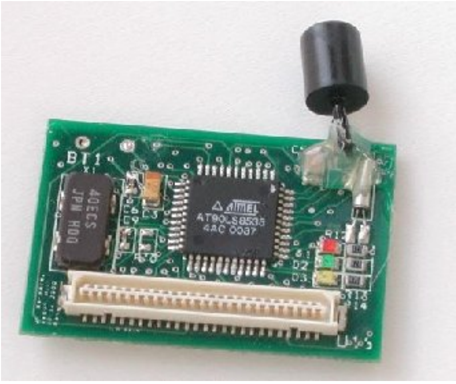
- Tiny, low-power, wireless sensors
- Minimal CPU, memory, and radio
  - Typically 8 Mhz CPU, 10 KB RAM
  - 100 m radio range, IEEE 802.15.4
- Extremely low power
  - A pair of AA batteries can power a mote for months or years!

TMote Sky
(Sentilla)



MicaZ (Crossbow)

WeC (Berkeley)



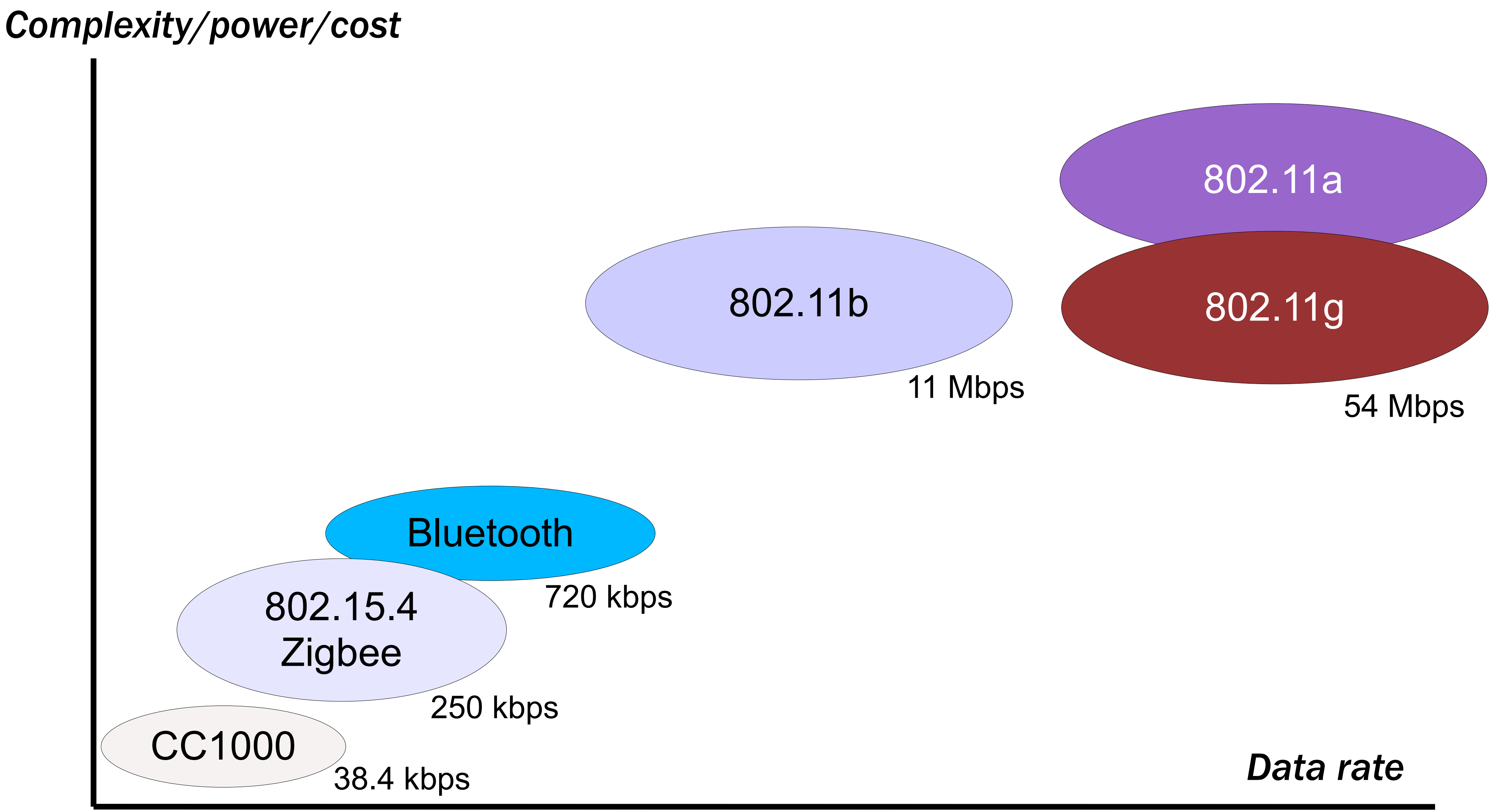
Rene (Berkeley)

iMote2 (Intel)

Key WSN Hardware Characteristics

- Limited CPU
  - Slow (8 MHz) -- No floating point computation.
  - 512-point FFT takes 450 ms, IFFT takes 144 ms.
- Limited memory
  - 10 KB of RAM and 60 KB of program ROM.
  - Much of this taken up by system software.
- Potentially lots of storage
  - Some designs support up to 2 GB of MicroSD flash
  - But, expensive to access: 13 ms to read/write a 512-byte block; ~ 25 mA.
- Low-power radio
  - 802.15.4 best case performance: 100 Kbps or so (single node transmitting, no interference, short range)
  - Approx 50 m range, and very unreliable!!

Wireless Technologies Comparison



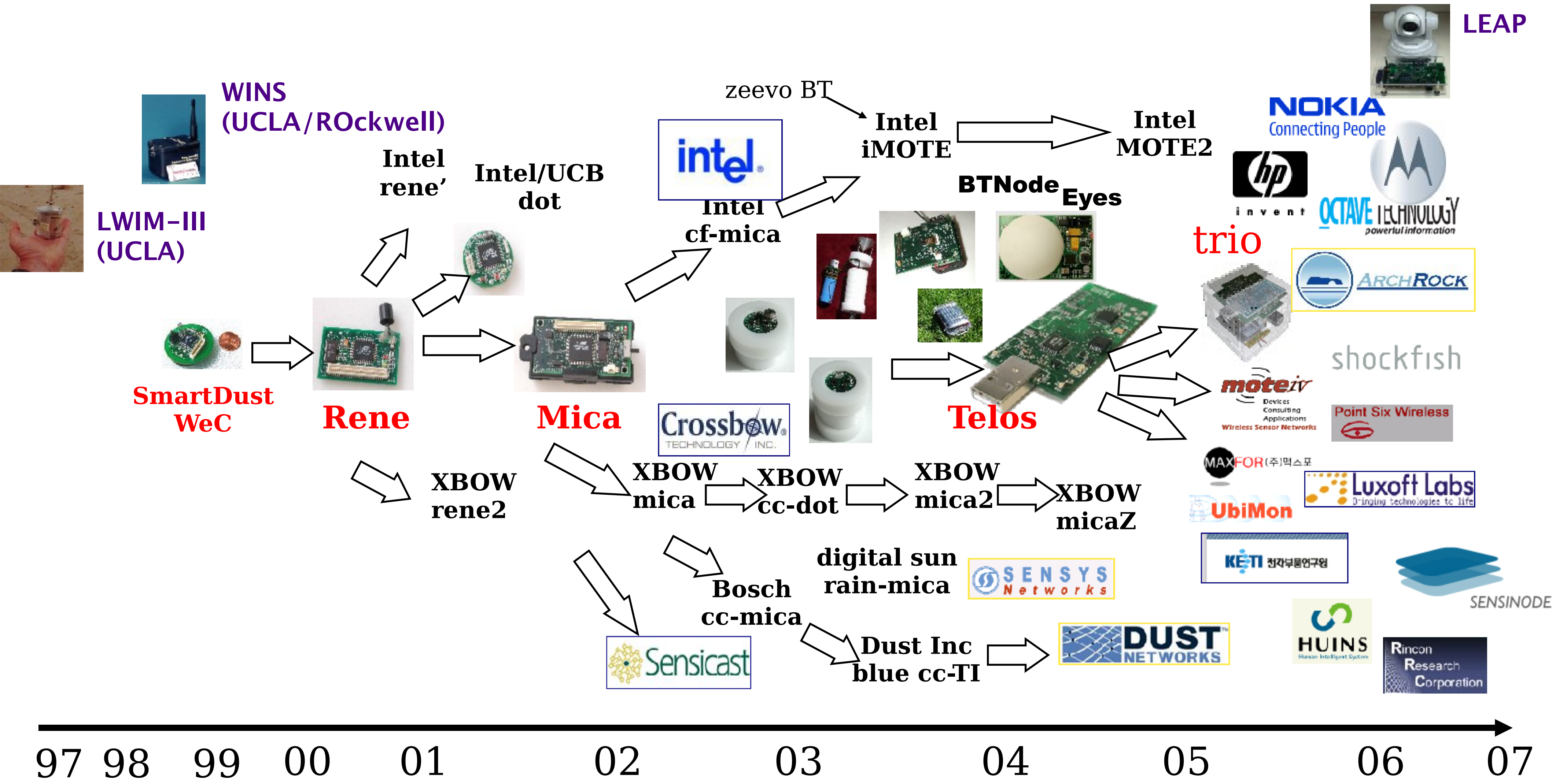
Wireless Technologies Comparison

Type	Data rate	Transmit pwr	Range (approx)	Cost
802.11b	11 Mbps	100 mW	100' – 300'	~$100
802.11g	54 Mbps	100 mW	< 802.11b	~$100
802.11a	54 Mbps	100 mW	80'	~$100
Bluetooth	720 kbps	1 mW / 30 mW	30' / 300'	~$5
802.15.4	250 kbps	1 mW	30 – 225'	~$5

Power Consumption

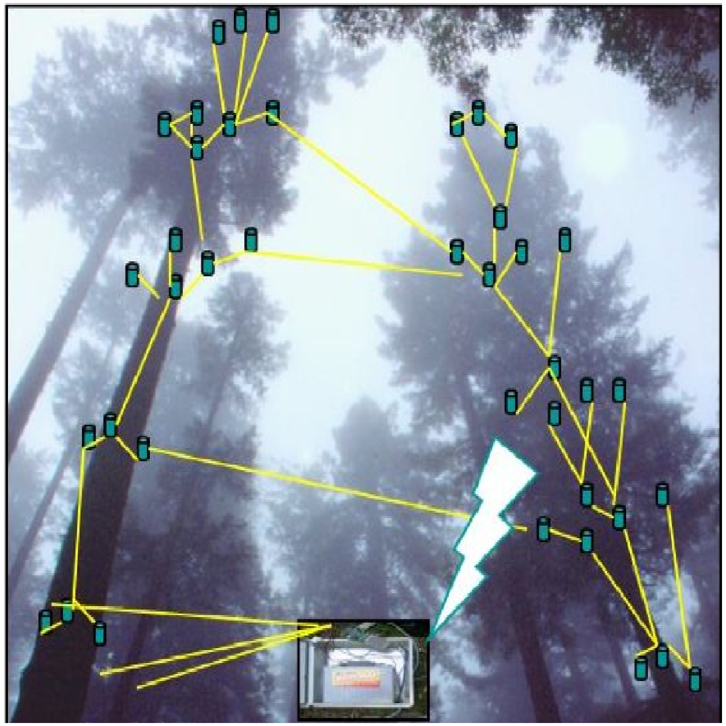
Type	*Current (receive)*	*Current (transmit)*
802.11b	170-350 mA	285-490 mA
Bluetooth	35 – 300mA active	35 – 300 mA active
802.15.4	19.7 mA	17.4 mA

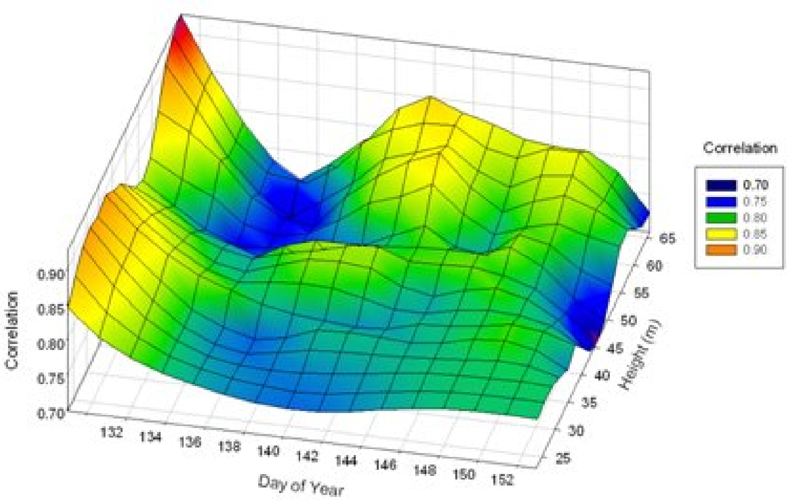
WSN Research Phenomenon…

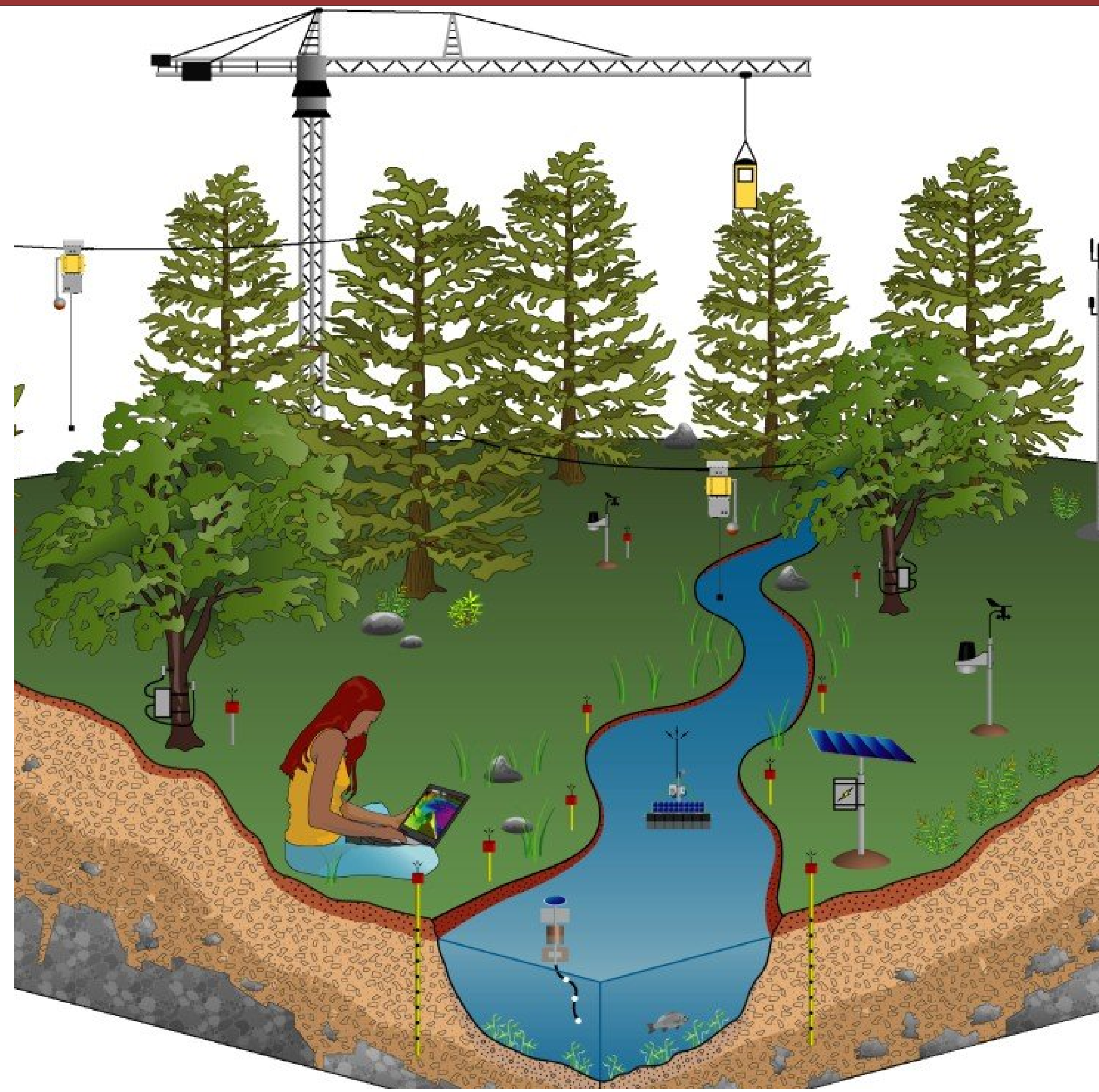


Environmental Monitoring

UCLA, UC Berkeley, many others

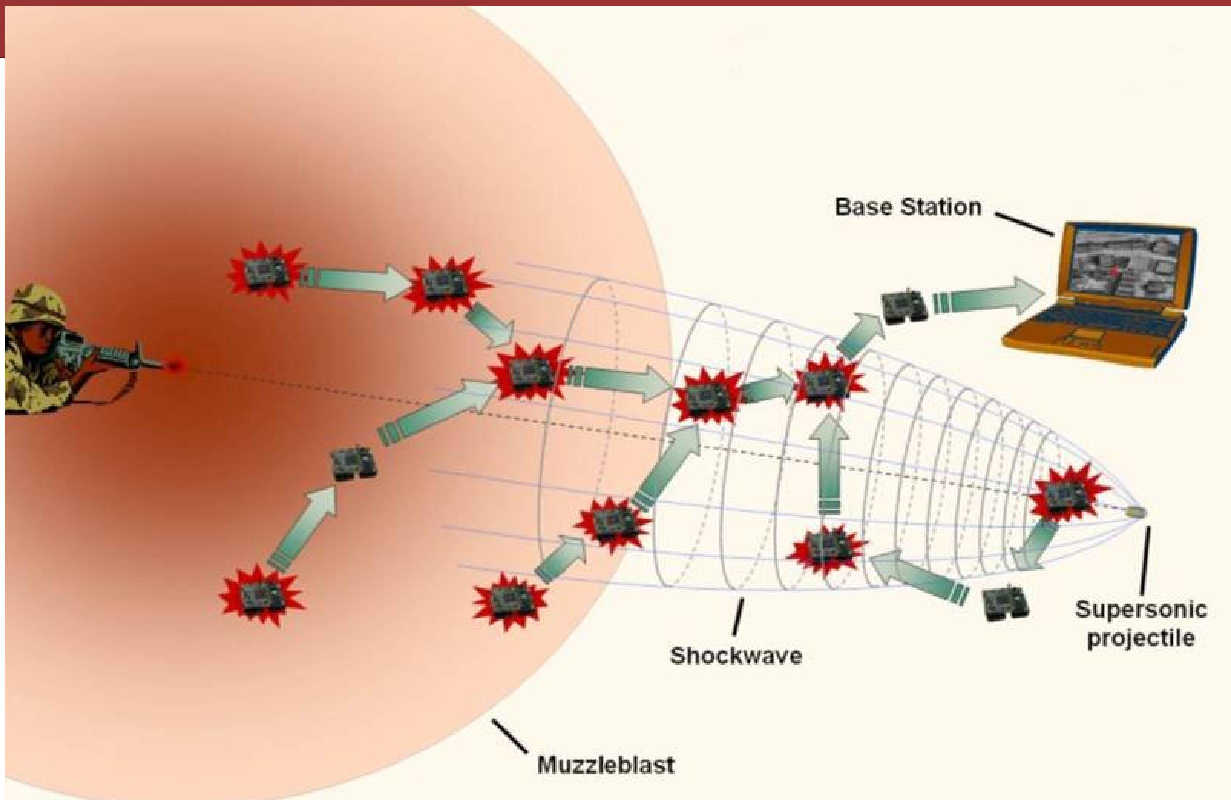


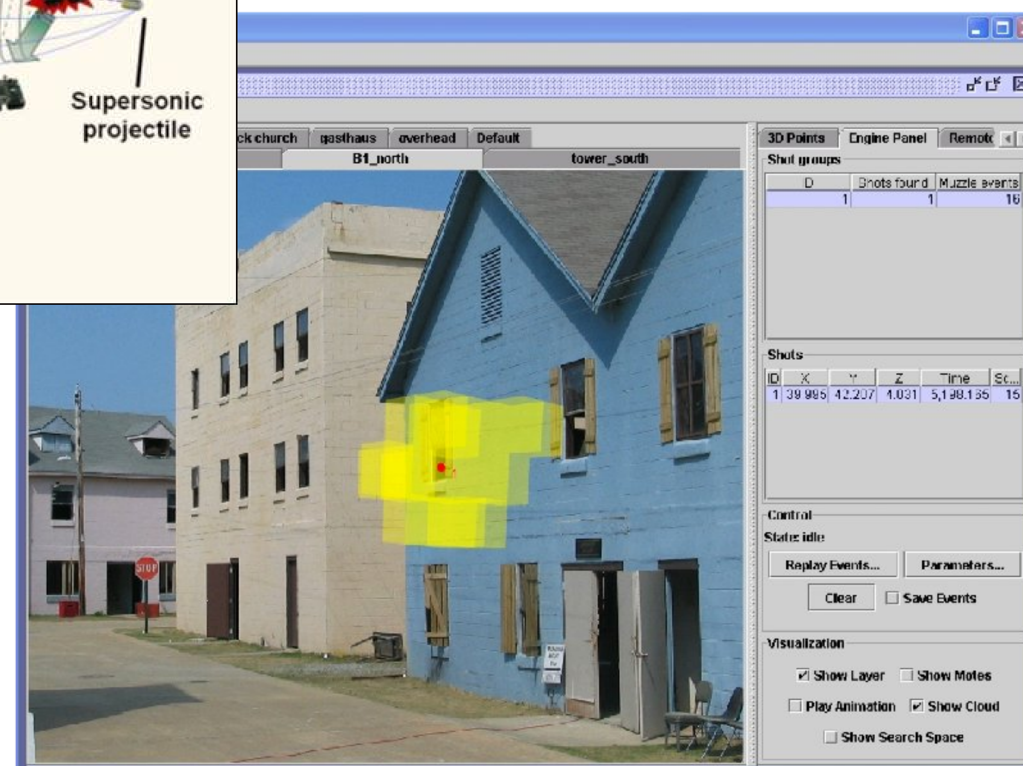




Gunshot Detection

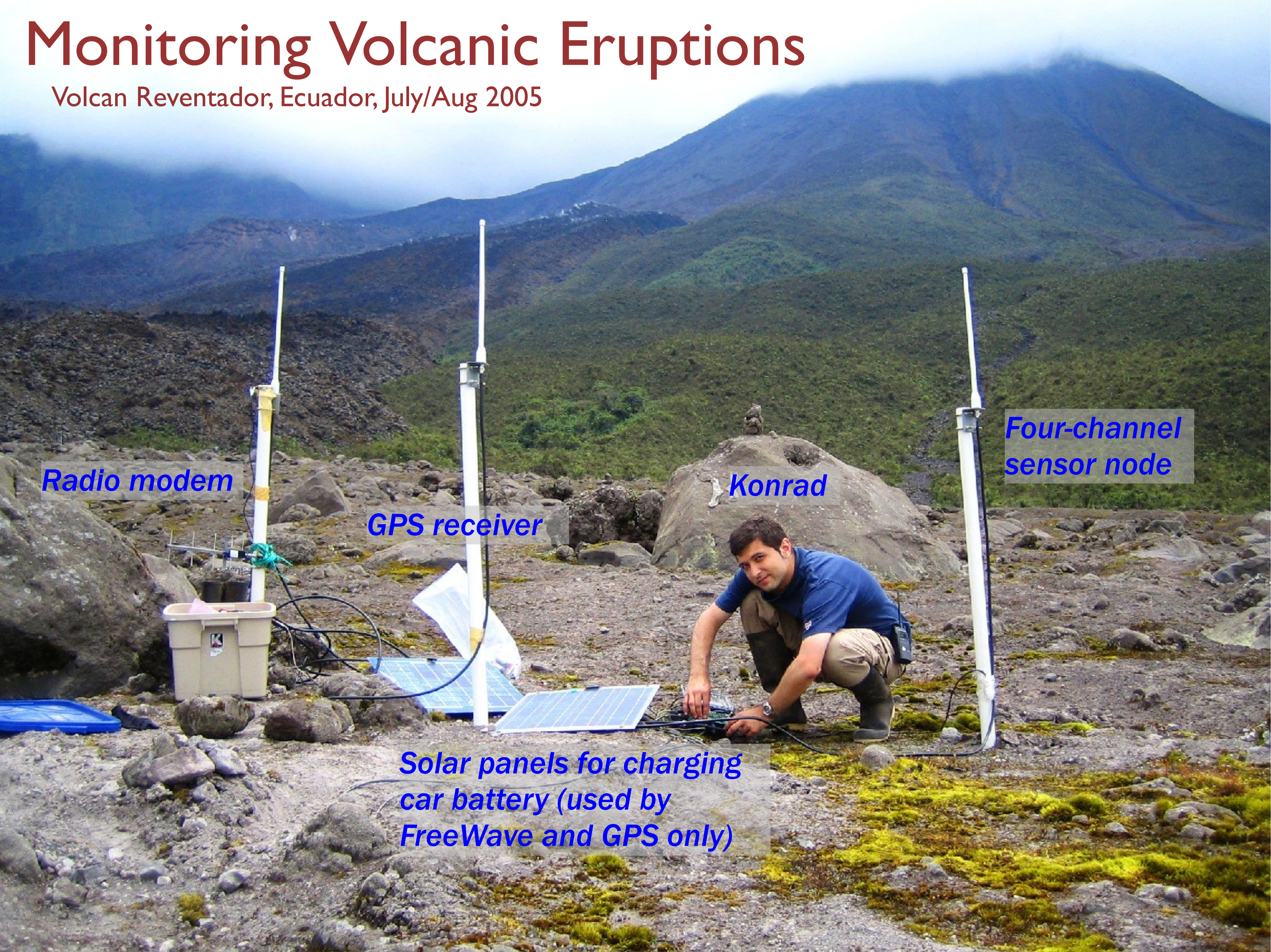
PinPtr, Vanderbilt





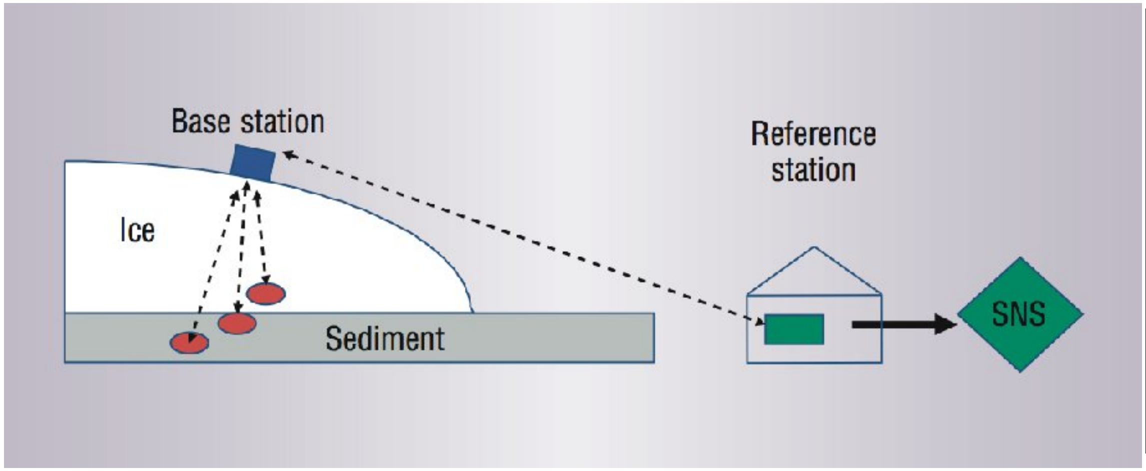
Monitoring Volcanic Eruptions

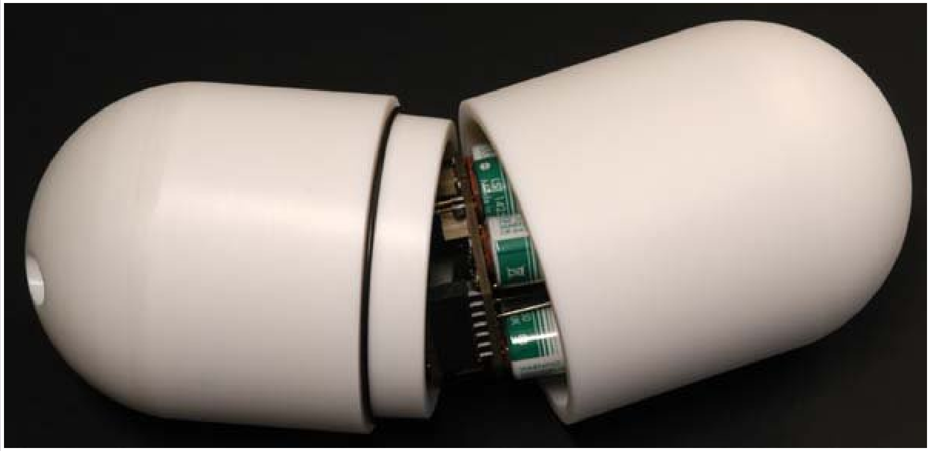
Volcan Reventador, Ecuador, July/Aug 2005



Glacier Monitoring

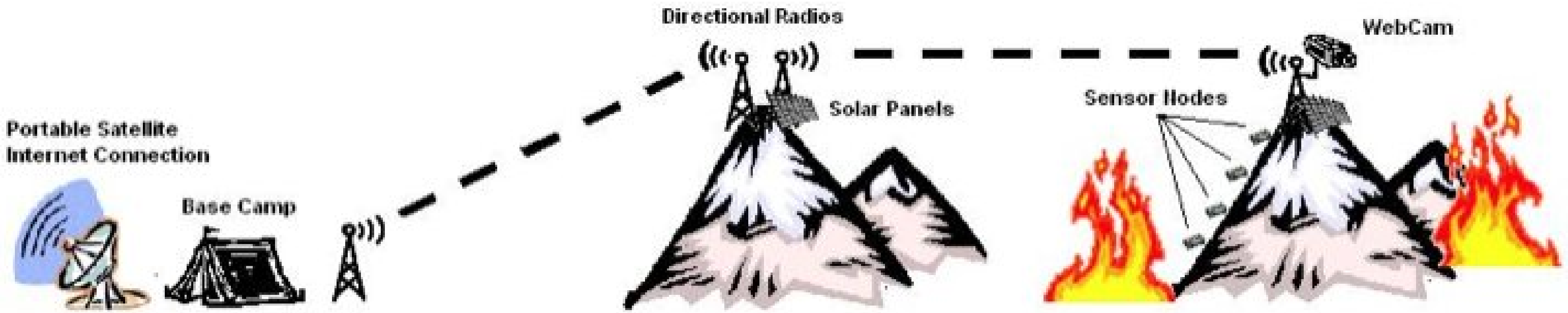
Glacsweb, Univ. Southampton





Forest Fire Detection

FireWxNet, Univ. Colorado

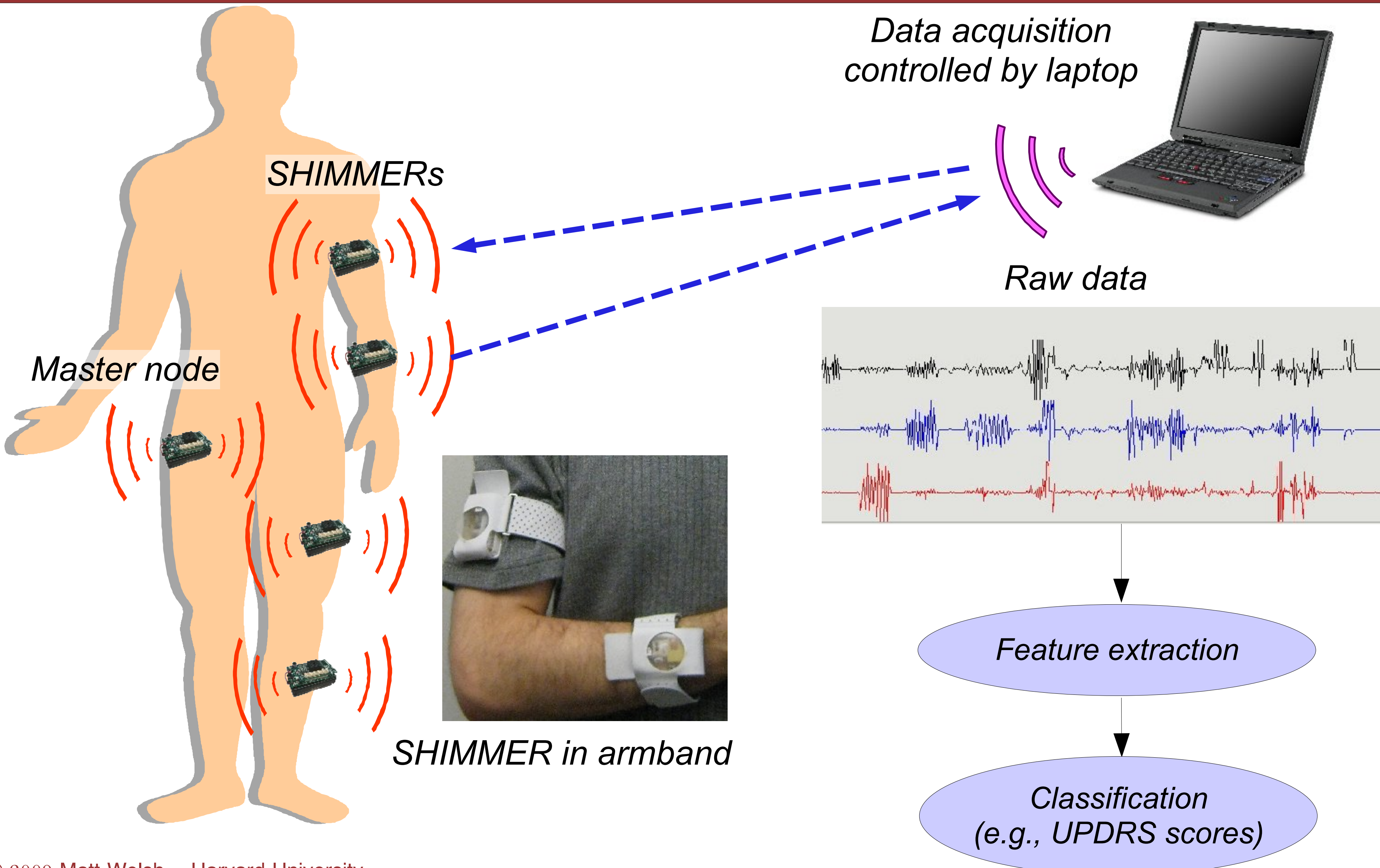


Emergency Medical Care and Disaster Response

CodeBlue, Harvard

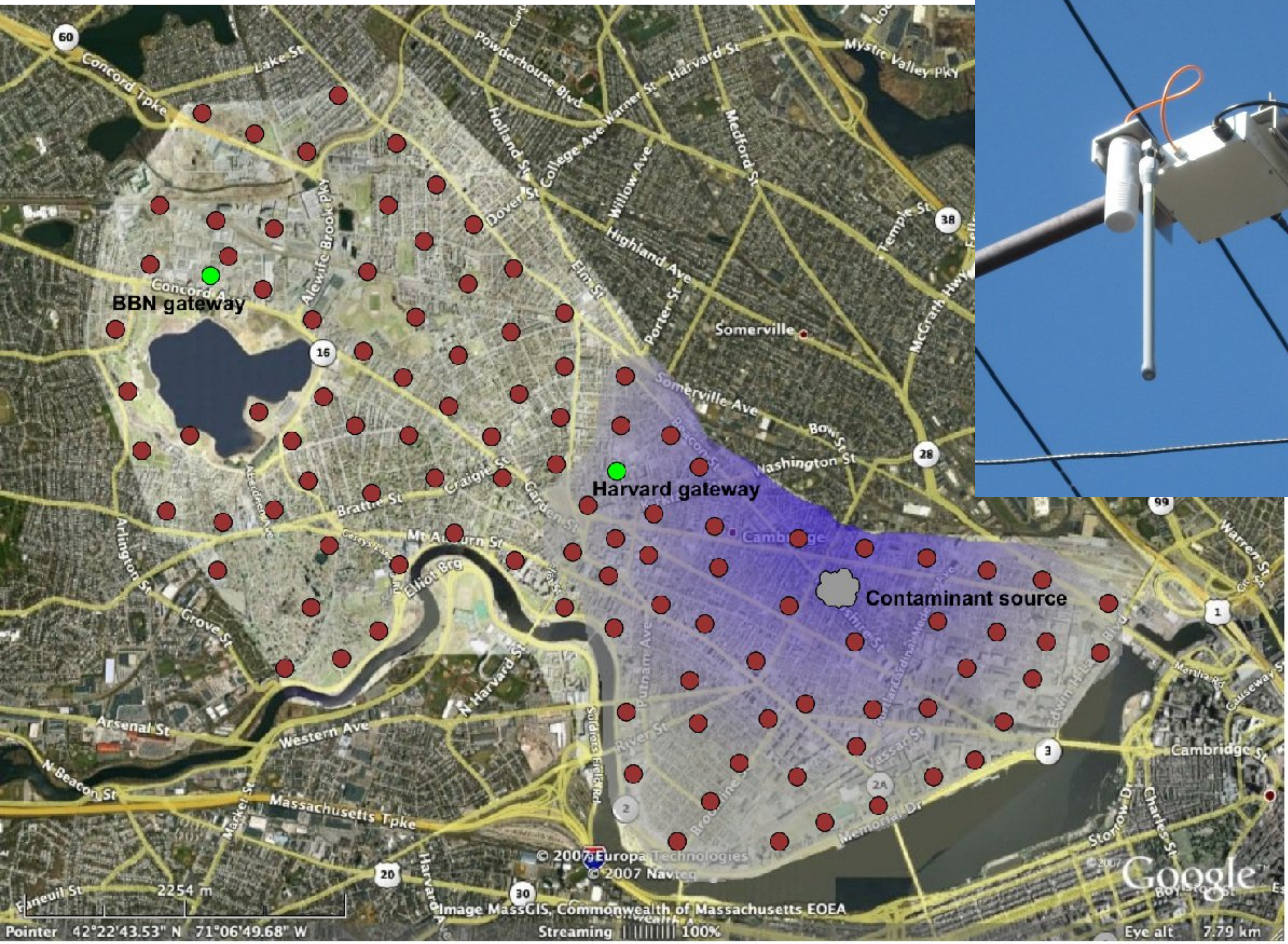
Neuromotor disease assessment

Mercury, Harvard



Urban-Scale Monitoring

CitySense, Harvard



The Macroscope

- *For the first time*, sensor networks allow us to:
- 1) Observe the world (environment, buildings, people, etc.) at very *high spatial resolutions*;
- 2) Make these observations *continuously*; and
- 3) Collect the observations in *digital form*.
    - Some have referred to this concept as a “macroscope” -- a scientific instrument that observes entire systems.

Intelligent Instrumentation

- Sensor networks are not just passive instruments!
- We can push processing and “intelligence” into the network.
- Processing can happen at many levels:
  - On individual sensor nodes.
  - At aggregation points within the network.
  - At the base station or gateway.
- Sensor networks fundamentally change the notion of “scientific observation” from a *passive* process to an *active* one.
  - This has a deep impact on many aspects of science.

Fundamental Research Questions

- Low-power wireless networking
  - Dealing with complexities of RF propagation – not a “disc model”
  - Limited bandwidth, very expensive to transmit, receive, and even listen!
  - Every node is a router – addressing, route selection, reliable transfers
- Operating system design
  - Motes have ~10 KB of RAM. Can't run Linux.
  - What are the right abstractions for concurrency, power management, communication?
- Distributed network services
  - Nodes in a WSN don't exist in isolation. They must coordinate their behavior.
  - Localization – how do you know where nodes are? Use RF signals? Ultrasound?
  - Time synchronization – how do nodes agree on a global clock?

Fundamental Research Questions (2)

- In-network sensor data processing
  - Communication is expensive: Sending one packet costs same energy as thousands of CPU cycles.
  - Always better to process the data closer to its source
  - Example: aggregation – nodes can collect data locally, compute aggregates (mean, max min, etc.) rather than sending raw data
  - Tracking: Sensors can collaborate to detect, localize, and track a target (tank or animal)
- Mobile, acoustic, and camera-based sensing
  - Very different sensing modalities and challenges
  - Acoustic and vision sensors require substantial computational horsepower
  - Mobile sensing involves (possibly unpredictable) variations in radio connectivity
  - How do we deal with noisy and intermittent measurements of the world?

What will this class be like?

This is a graduate research seminar.

- We will mostly be reading and discussing research papers
- Roughly 4 papers a week

Prerequisites:

- Must either be a CS grad student **or** have taken either CS161 or CS143.
- Must feel comfortable programming in C.

One programming assignment

- Introduce you to programming sensor networks using the Pixie OS and NesC language
- Run on the Harvard MoteLab sensor network testbed

Research project

- You pick the topic, write a proposal, do the project, give presentation, write final report

Readings and Reviews

- You are responsible for completing assigned readings *before* lecture
  - Usually 2 papers for each class
- Email a short review of the reading to cs263-staff@eecs
  - Review is due before beginning of lecture
  - A couple of paragraphs about the reading
  - Highlight the main "take away" point of the reading
  - Provide a short critique of the work as well
    - *Be concise, critical, and thoughtful*
- Reviews constitute 25% of your course grade
  - You are allowed to miss two classes of paper reviews over the term

Course Blog

- http://harvard-cs263.blogspot.com
- Blogging class discussions
  - Each class, one person will blog the discussion and post it later that day
  - You are welcome to post comments, thoughts, musings, etc. as comments
  - Or, you can blog anything else you want (related to the course material).
  - This blog is public so be technically accurate and respectful!

Programming Assignment

- There is one programming assignment for the course
  - Main goal: Get experience programming a real sensor network
  - You will use this experience for your course project
- Project will involve designing a multihop routing protocol, running on the Pixie OS, on the Harvard MoteLab sensor testbed
- You should be comfortable programming in C

Research Project

- Main goal of this course: Do some research
  - Work individually or in pairs (pairs preferred)
  - Select a juicy research problem that fits the theme of this course
- Use the project to further your own research goals
  - Ideal project is one that fits in with your own thesis topic in some way
  - Focus of project need not be on “systems” and “networks”
    - *e.g., theory, AI, languages, hardware design, etc. are all valid*
    - *As long as it ties into the course topic in some way*

Project Requirements

- Project Proposal
  - Short (4 pages max) on what you propose to do, why the project is interesting, and how you plan to get started
  - Should include rough schedule of project milestones
  - Short project update due midway through semester – short email on where you are and how you plan to finish up your project
- Research presentations (last two days of class)
  - Give a short, fun talk telling us what you did
  - Learn from each other's experiences
- Research papers
  - Conference-style research paper (12 pages max) detailing your project
  - Goal is to get used to writing these things – it's important
  - I can work with you afterwards to to turn it into a conference/journal submission

Project Ideas

- Develop an adaptive time-sync protocol that tunes packet transmission rates based on energy availability
- Develop a sensor duty-cycling algorithm that accounts for energy drain and energy collect (e.g., using solar panels)
- Develop a tool to characterize and visualize energy and bandwidth consumption across a sensor network, use to identify hotspots and load imbalance
- Design a new sensor scripting language that includes resource constraints as a primitive
- Develop a technique to automatically detect and diagnose software and communication failures in a sensor network

Course staff and administrivia

- Instructor: Matt Welsh (mdw@eecs)
  - Office: Maxwell Dworkin 233
  - Office hours: Thursdays, 10am – 12pm
- TF: Bor-rong Chen (brchen@eecs)
  - Office: Maxwell Dworkin 238
  - Office hours: TBD
  - General course consulting and help with programming assignment
- All papers, due dates, etc. on course web page:
  - http://www.eecs.harvard.edu/~mdw/course/cs263/

Syllabus

- http://www.eecs.harvard.edu/~mdw/course/cs263
- Primarily research papers from the last few years of key conferences in the area: SenSys and IPSN in particular.
- Most papers about 14 pages in length.

Other Policies

- Enrollment will be limited to 15 students
  - Preference given to grad students in CS, then grad students in other disciplines, then undergrads in CS, then undergrads in other disciplines.
- No laptops!
  - Unless you are blogging that week.
- No pass/fail grading option for this course.

Grading

- 25% - Class participation and discussion
  - Come to class, participate in the discussion, ask questions, speak up!
- 25% - Paper summaries
  - Allowed to miss two days' worth of summaries during the term
- 10% - Programming assignment
- 40% - Final project
  - Graded on original proposal, final report, and in-class presentation

Next lecture

- Two papers to read for next lecture:
- System architecture directions for networked sensors
  - Jason Hill et al., ASPLOS 2000
- Analysis of a Large Scale Habitat Monitoring Application
  - Robert Szewczyk, SenSys 2004
- Send reviews to cs263-staff@eecs before class!
- Come prepared to talk!!!