Design Tools for Sensor-Based Science

Randal Burns
Andreas Terzis
Department of Computer Science
The Johns Hopkins University

Michael J. Franklin
Department of Electrical Engineering and Computer Science
University of California Berkeley

A High-Level Toolkit

- Deploying a sensor-based experiment is tedious
  - primitive techniques
  - complex systems: distributed and heterogeneous
  - variable and unpredictable
- Trial and error configuration damages sites
  - find rapid solutions – in number of iterations
- Requires substantial time and expertise

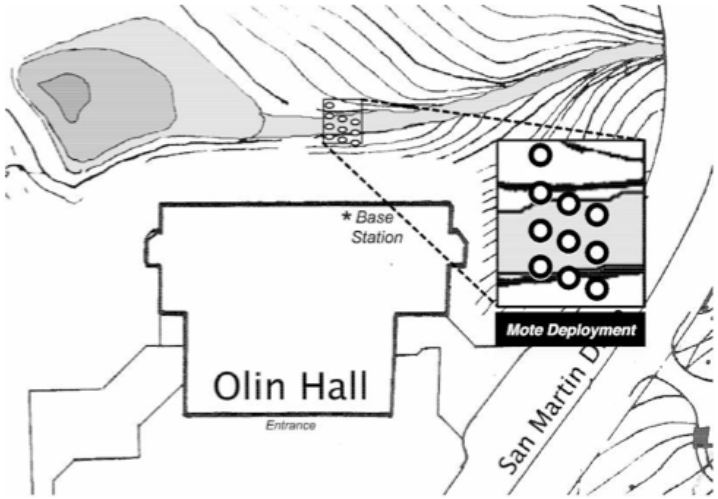
- *Critical* need for automated design tools
  - interfaces designed for the scientist

System Goals

- Reduce deployment complexity
  - automatically configure and program an end-to-end data collection and analysis application
  - motes, in-network processing, database schema, datacube
- Simple, intuitive tools to explore tradeoffs among cost, reliability, and network lifetime
  - explore what-if design scenarios prior to deployment
  - no manual tuning of placement or network parameters
- A “design wizard” for sensor-based science

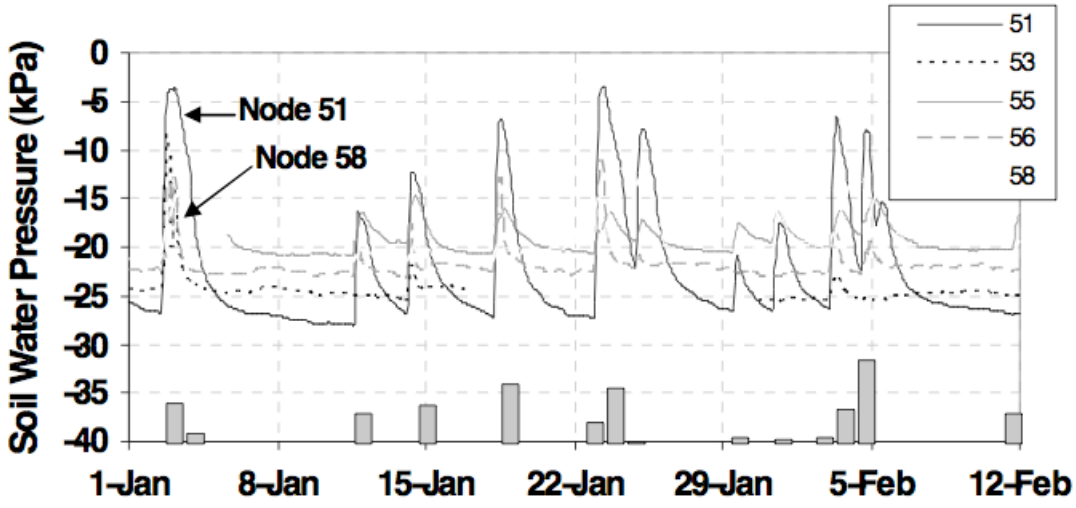
A Soil-Ecology Network

- Meso-scale monitoring of soil-ecology
  - 10 motes in an irregular grid with 3m spacing
  - fills a scientific gap between laboratory (bucket scale) and manual sampling
  - continuous monitoring tracks punctuated events and trends
- Semi-urban, woodland site
  - stream and surface water
  - WiFi and EM interference



Sample Results

- Continuous monitoring
  - temporal scale is most scientifically unique
- Integrates external data sources
  - rainfall at BWI (~10 miles away)
- Fine-grained variability within a 10 m grid



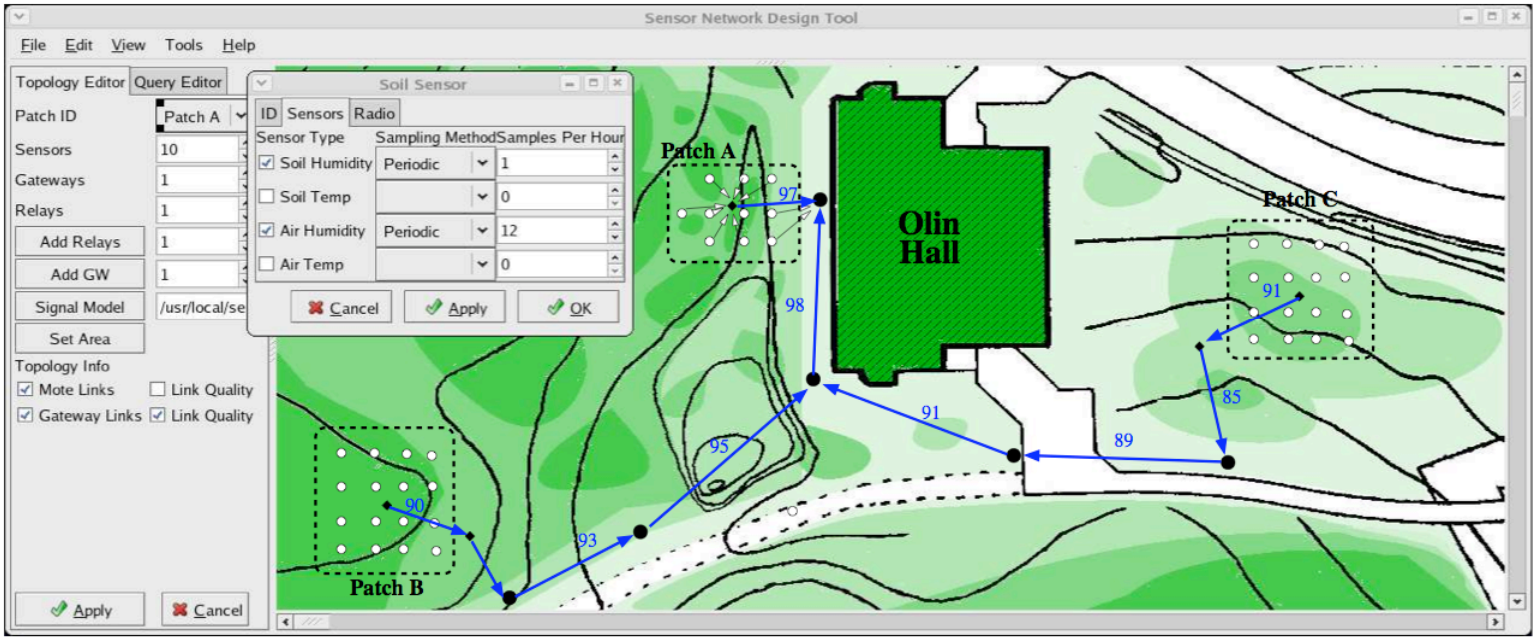
Motivation: Our Experience

Deploying a sensor network for site-monitoring:

- Is labor intensive
  - 400 person hours including 80 faculty hours
- Demands expertise
  - 5 computer scientists and 2 physicists for every ecologist
  - programming heterogeneous, distributed systems
- Uses custom components
  - soldering, device drivers, reliable transfer protocol
- Requires manual tuning
  - just to connect to all motes

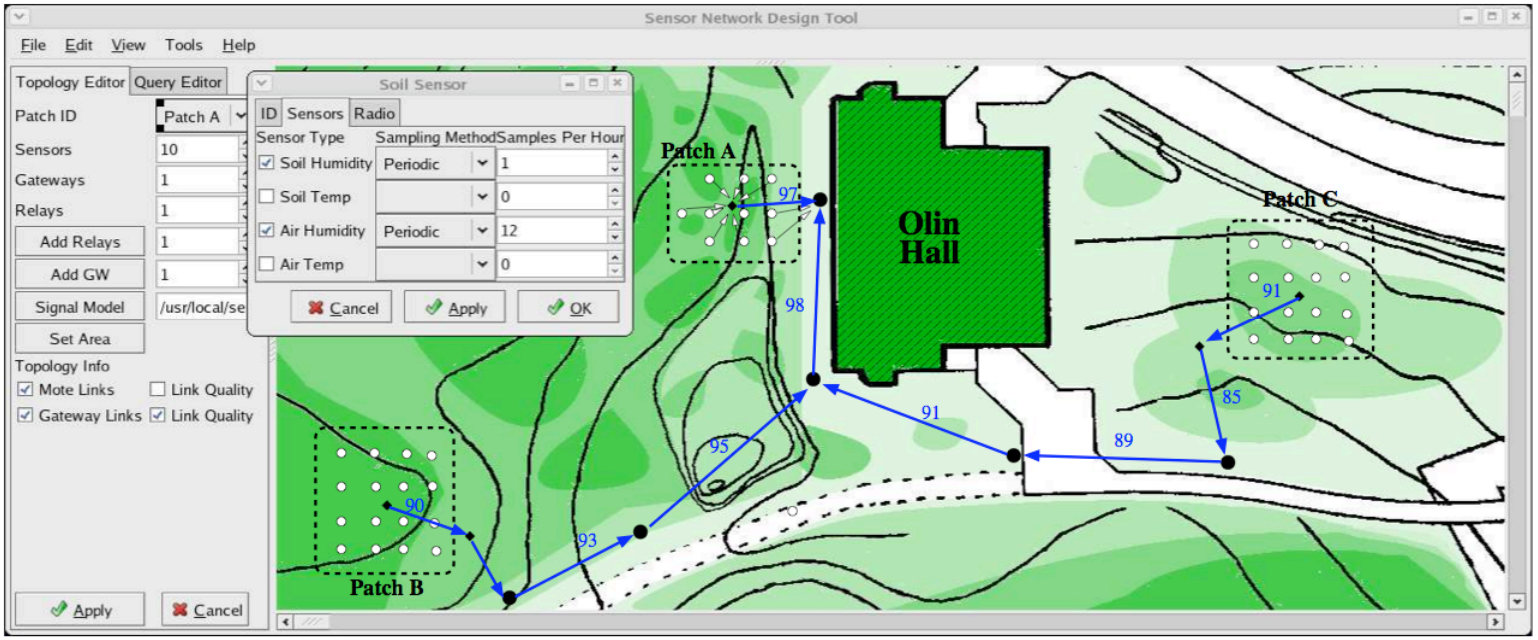
Cartographic Interface

- Drag and drop hardware to monitoring sites on a map
  - configure sampling discipline
- Annotate transmission properties and obstacles
- Write simple (excel-like) equations



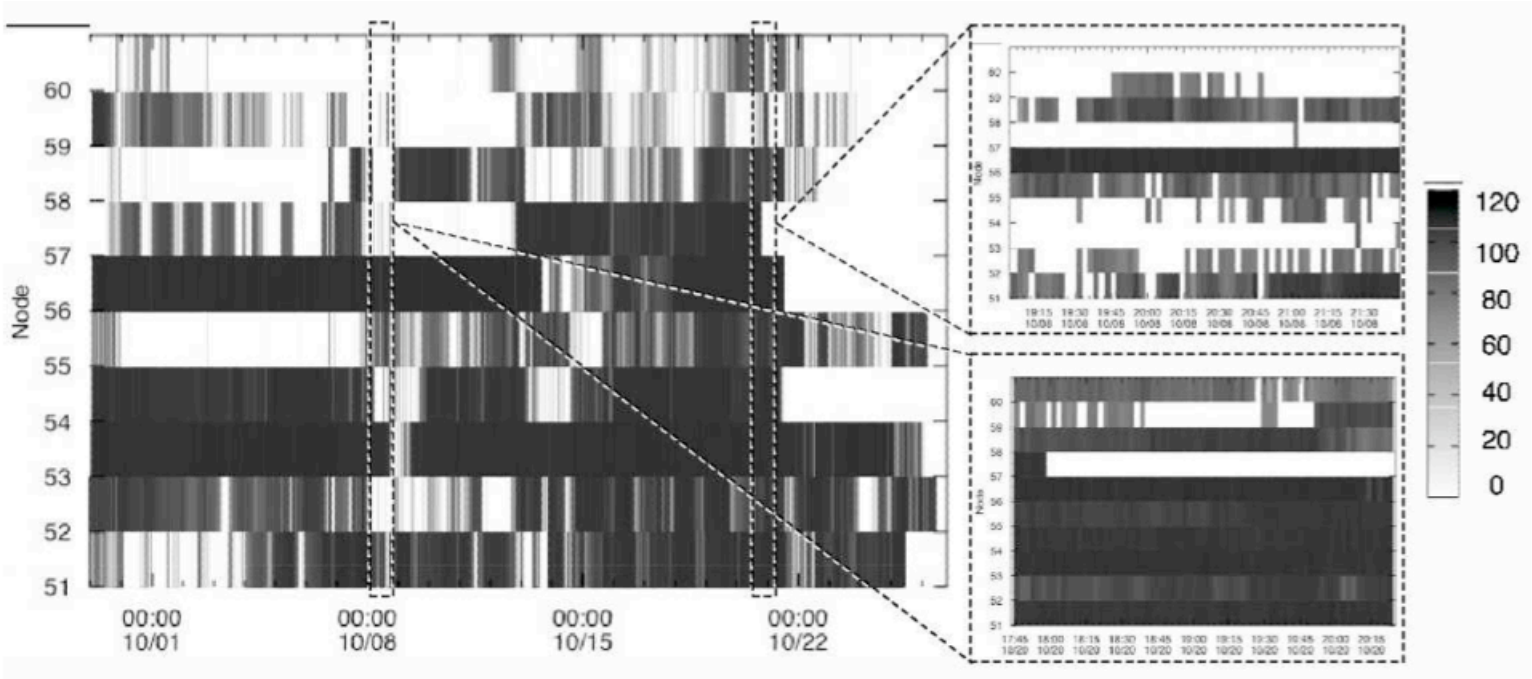
Ask for a Solution?!

- Enhanced network with relays and gateways
  - subject to a hardware budget
- Availability and lifetime properties of network
  - annotations of link properties



Link Quality Indicators

- Network link performance
  - is “relatively” unpredictable
  - varies temporally and spatially



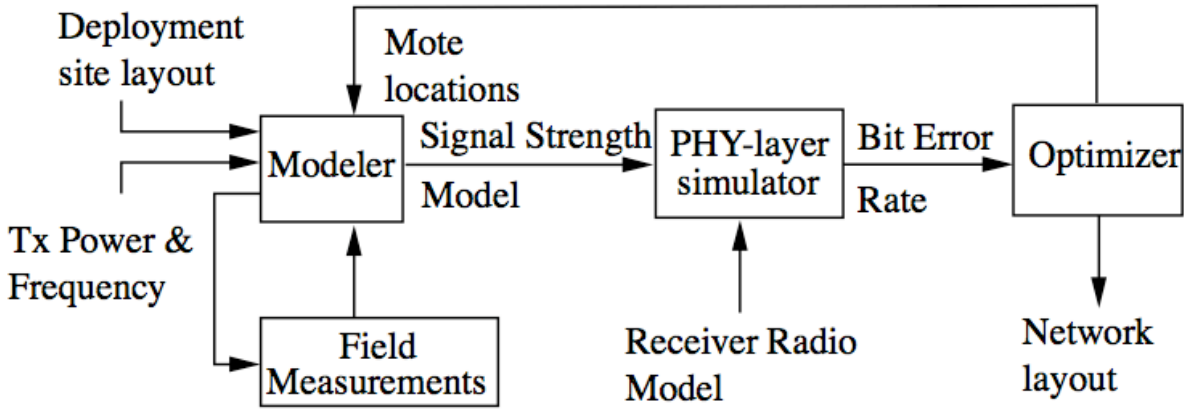
Network Design

- Requires site-specific signal propagation model
  - connectivity among nodes and its time variance
- Iterative refinement
  - of existing network models over time
  - from initial networks to subsequent deployments
- Take user/expert input
  - obstacles: from an obstacle library
  - annotate clear paths and regions of attenuation

- Goal: construct a *practical,* experimental model
  - purely analytical modeling has complexity and applicability concerns

Network Design Tool

- Inputs
  - Site information: size, layout, type, obstacles
  - Existing network model from previous deployment
  - Hardware (allocated and unallocated) and properties
- Outputs
  - topology: augmented network with relays and gateways
  - network quality: least-lossy tree



Finding Solutions Rapidly

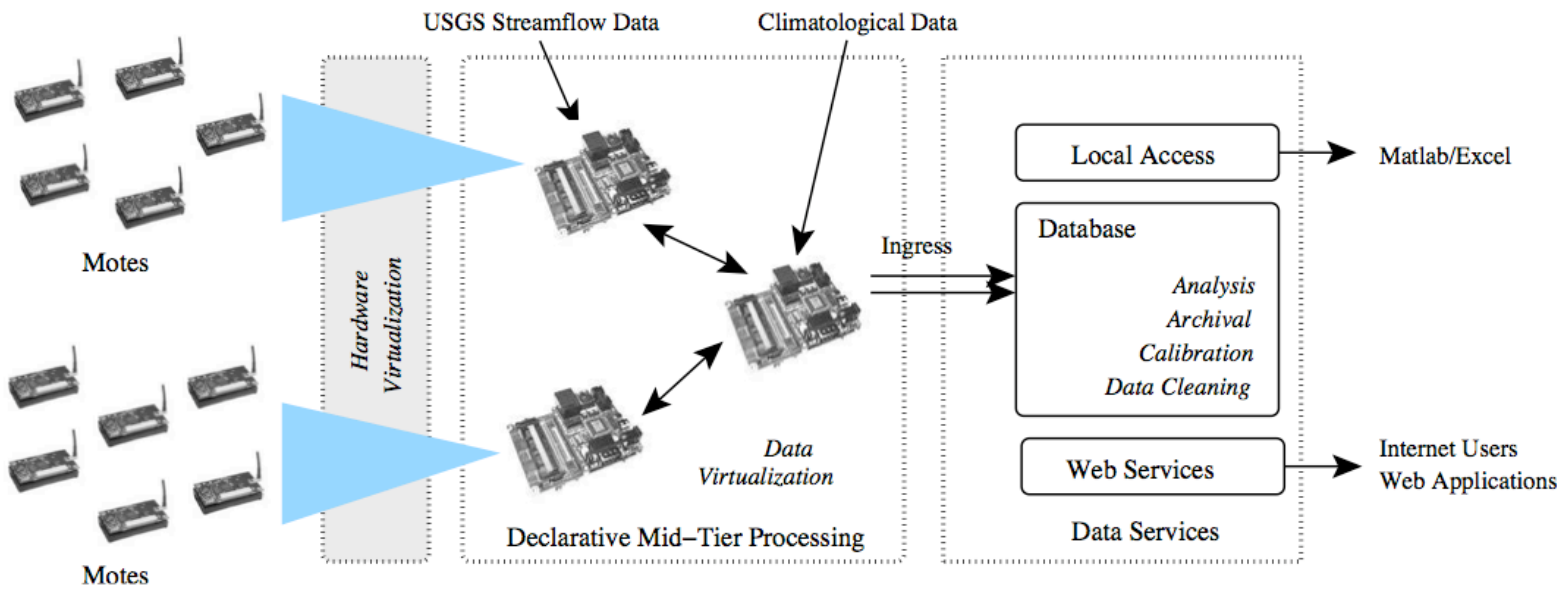
- Many sites are fragile and hostile to hardware
  - high-variance in performance metrics
  - frequent failures
- Avoid damaging site incursions
  - from trial and error configuration and repair
- Design-based deployment strategies
  - a small network to “learn” site characteristics
    - condition transmission models
  - followed by a full-scale deployment
  - achieve a solution rapidly and accurately
    - minimally overprovision in few iterations

Data Design

- Data-driven application programming
- Automatic construction of:
  - database schema
  - analysis tools, *e.g.* datacube
  - network data handling: stream processing
  - integration of external data sources
  - Interfaces: forms and Web-services
- Two-levels of data processing
  - Declarative mid-tier processing (on motes and gateways)
    - process inbound data and populate the database
  - Data services: store and analyze data

Architecture

- Layers of network and data virtualization
- Redundant function in mid-tier and data services
  - data services are archival

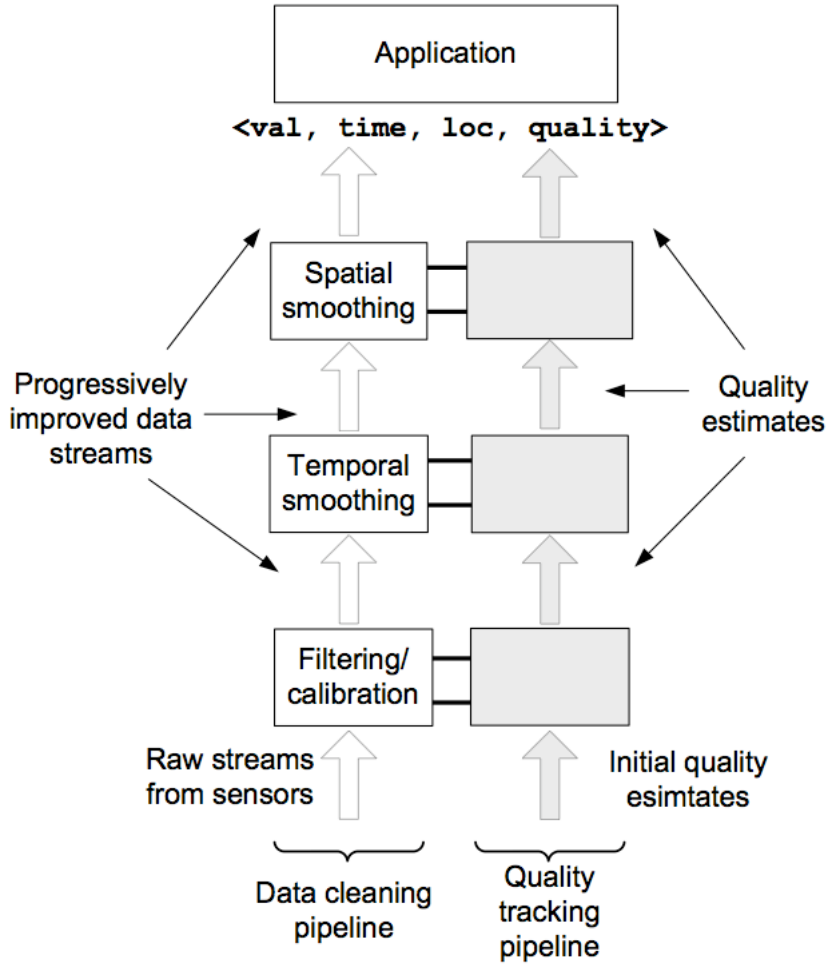


Declarative Mid-Tier Processing

- Built on HiFi data processing system
  - focused on self-management and ease-of-use aspects in error-prone environments
- Stream processing software on gateway devices
  - computes aggregates and filters
  - correlates/merges multiple data streams
- Benefits of a declarative approach
  - automatic and dynamic query optimization
    - including query reuse, *i.e.* result caching
  - incorporate a wide range of data sources

Data Cleaning

- Discrepancies between physical and digital world
  - inherent ambiguities and inconsistencies
  - variance and unreliability of data acquisition devices
- Virtual Devices
  - data pipeline
  - fuse results from multiple sensors
- Notions of answer quality:
  - error bounds
  - confidence intervals



Concluding Remarks

- Many difficult problems
  - network provisioning solves the *Facility Location* problem
  - automated generation of data analysis
- Eliminate barriers to sensor deployments
  - make sensors accessible to ecologists (and others)
  - unlock the potential of sensor networks
- Evaluation plan
  - growing our network to 200 motes at 2 Baltimore sites