CitySense

An Open, City-Wide Wireless Sensor Network

Matt Welsh

Harvard University

School of Engineering and Applied Sciences

Josh Bers

BBN, Inc.

CitySense Concept

100+ Linux PCs with 802.11a/b/g
on buildings and streetlights throughout a city
(current target is Cambridge, MA)

Sensors for monitoring air quality, weather,
road traffic, contaminants, ...

The network is programmable
by anyone

Why CitySense?

Expand sensor networking testbeds beyond indoor deployments with resource-constrained nodes

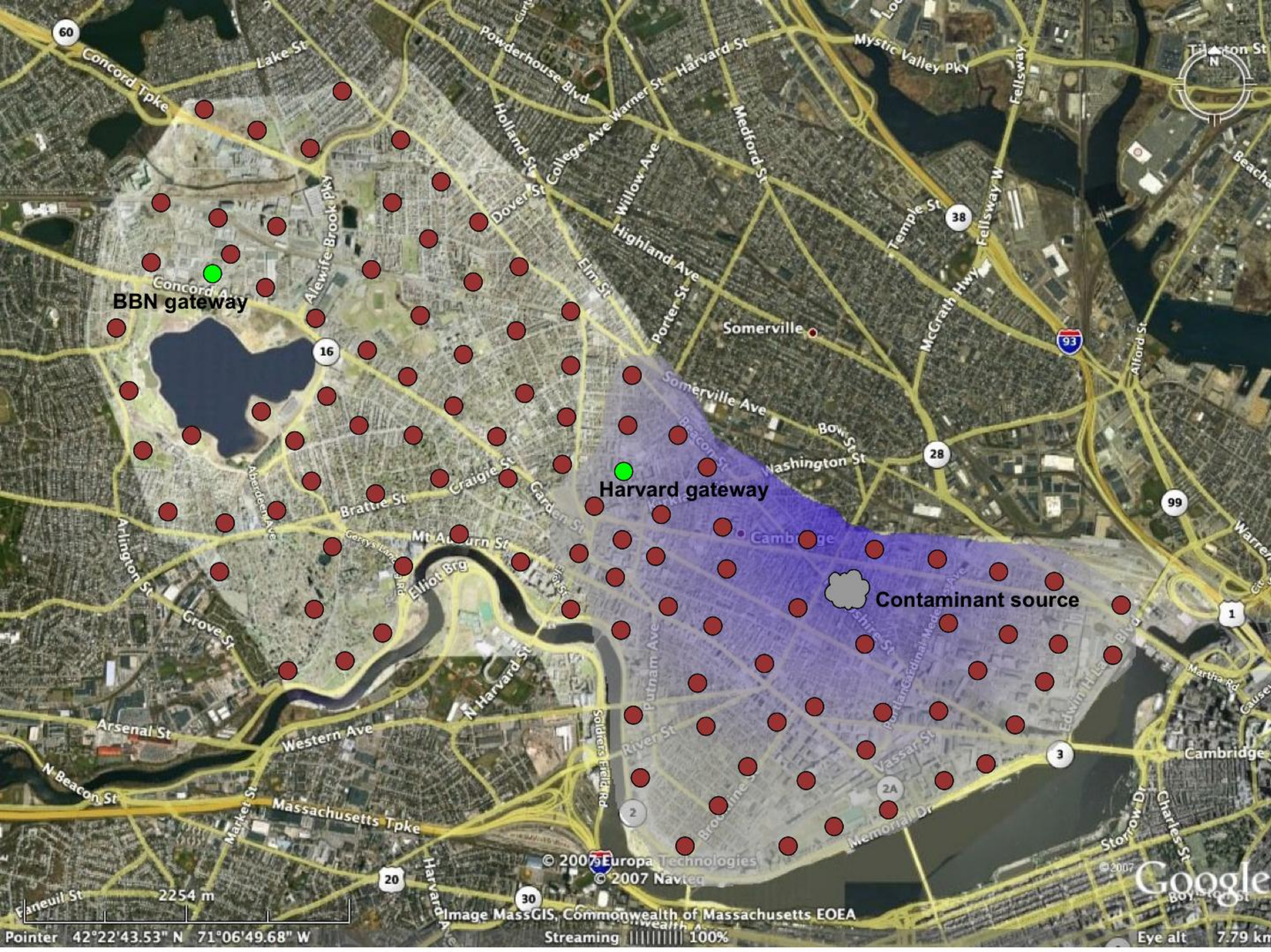
- Outdoor testbed with large coverage area
- Powered nodes with substantial CPU/memory/radio bandwidth
- Provide blueprint for future sensor network designs and deployments

Shared resource open to research community

- Leverage experience with Harvard's MoteLab to provide shared experimental facility

Provide bridge to broader scientific communities

- Partnership with Harvard School of Public Health
- NSF GENI initiative
- Educational impact at graduate, undergraduate, and K-12 levels


BBN gateway
Harvard gateway
Contaminant source
Somerville
Cambridge
Concord Tpke
Lake St
Powderhouse Blvd
Harvard St
Mystic Valley Pky
Fellsway
Holland St
College Ave
Warner St
Willow Ave
Medford St
Dover St
Elm St
Highland Ave
Porter St
Temple St
Fellsway W
McGrath Hwy
Alewife Brook Pky
Somerville Ave
Bow St
Washington St
Alford St
Craigie St
Garden St
Brattle St
Aberdeen Ave
Arlington St
Grove St
Mt Auburn St
Elliot Brg
Putnam Ave
River St
Vassar St
Memorial Dr
Edwin H Land Blvd
Martha Rd
Storrow Dr
Charles St
Soldiers Field Rd
N Harvard St
Western Ave
Arsenal St
N Beacon St
Market St
Massachusetts Tpke
Faneuil St
2254 m
© 2007 Europa Technologies
© 2007 Navteq
Image MassGIS, Commonwealth of Massachusetts EOEA
Google
Pointer 42°22'43.53" N 71°06'49.68" W
Streaming 100%
Eye alt 7.79 km

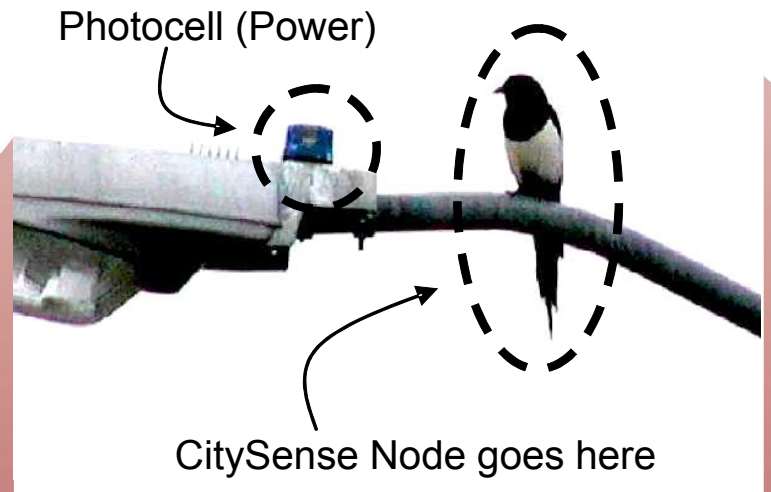
CitySense Node Design

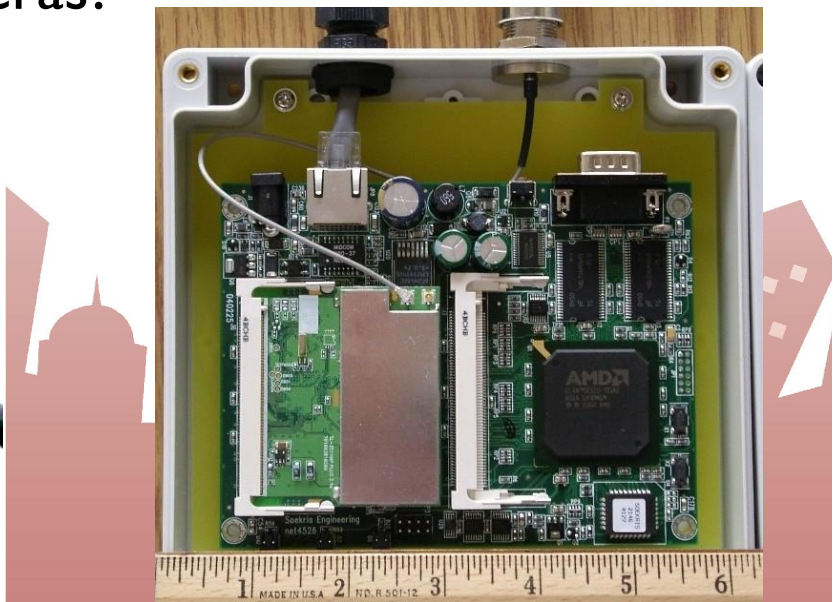
Soekris net4826 embedded PC running Linux

- 256 MB of RAM+flash, 1 GB USB flash drive
- Two Atheros 802.11a/b/g radios, 8.5 dBi omni antennas

Various sensors driven by app requirements

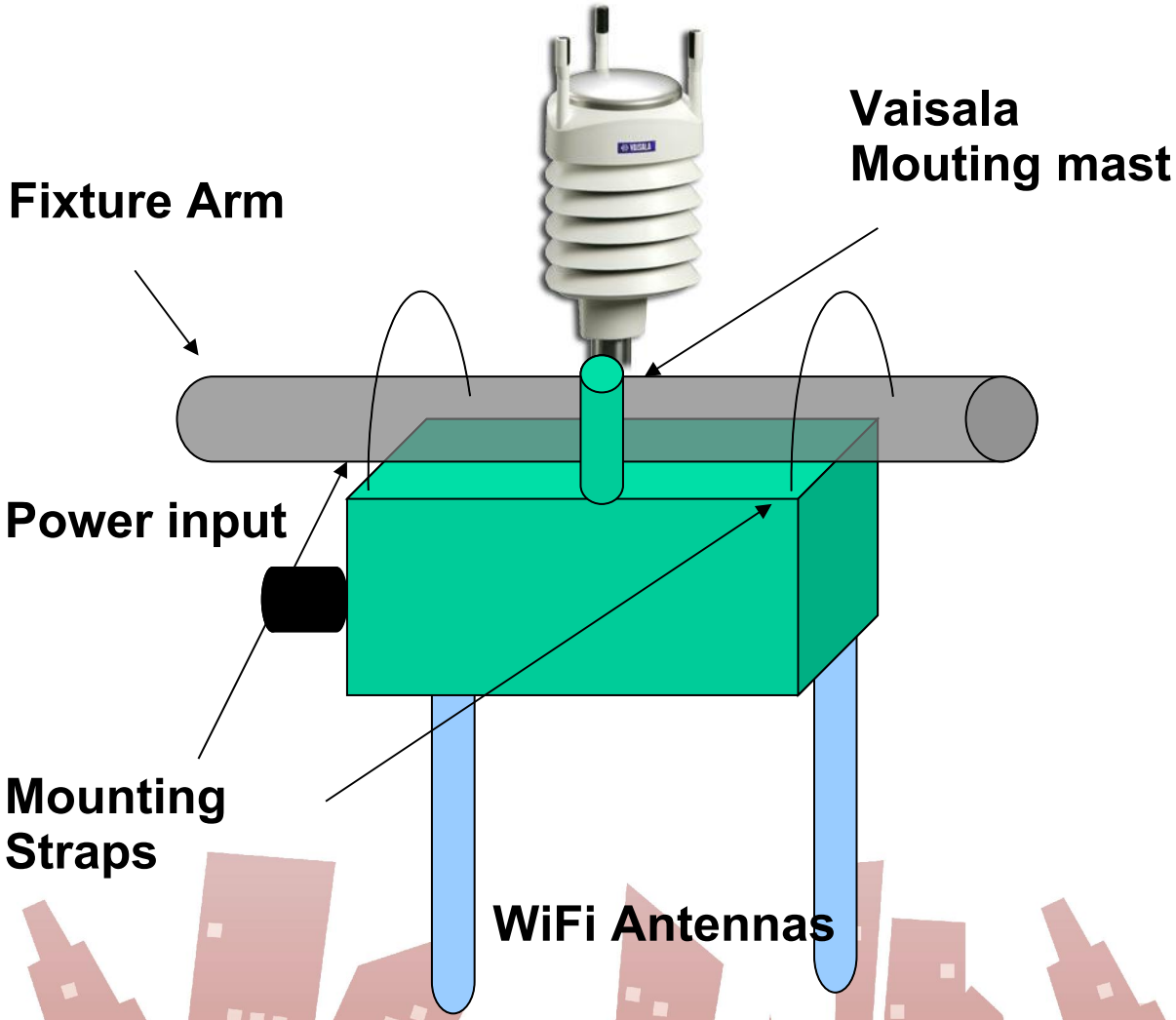
- Vaisala Weather Sensor (temp, humidity, pressure, wind, rain, ...)
- Air particulate sensor (PM10 for air quality studies)
- Gas sensors (CO_2, CO, NOX, ...)
- ~~Microphones? Cameras?~~





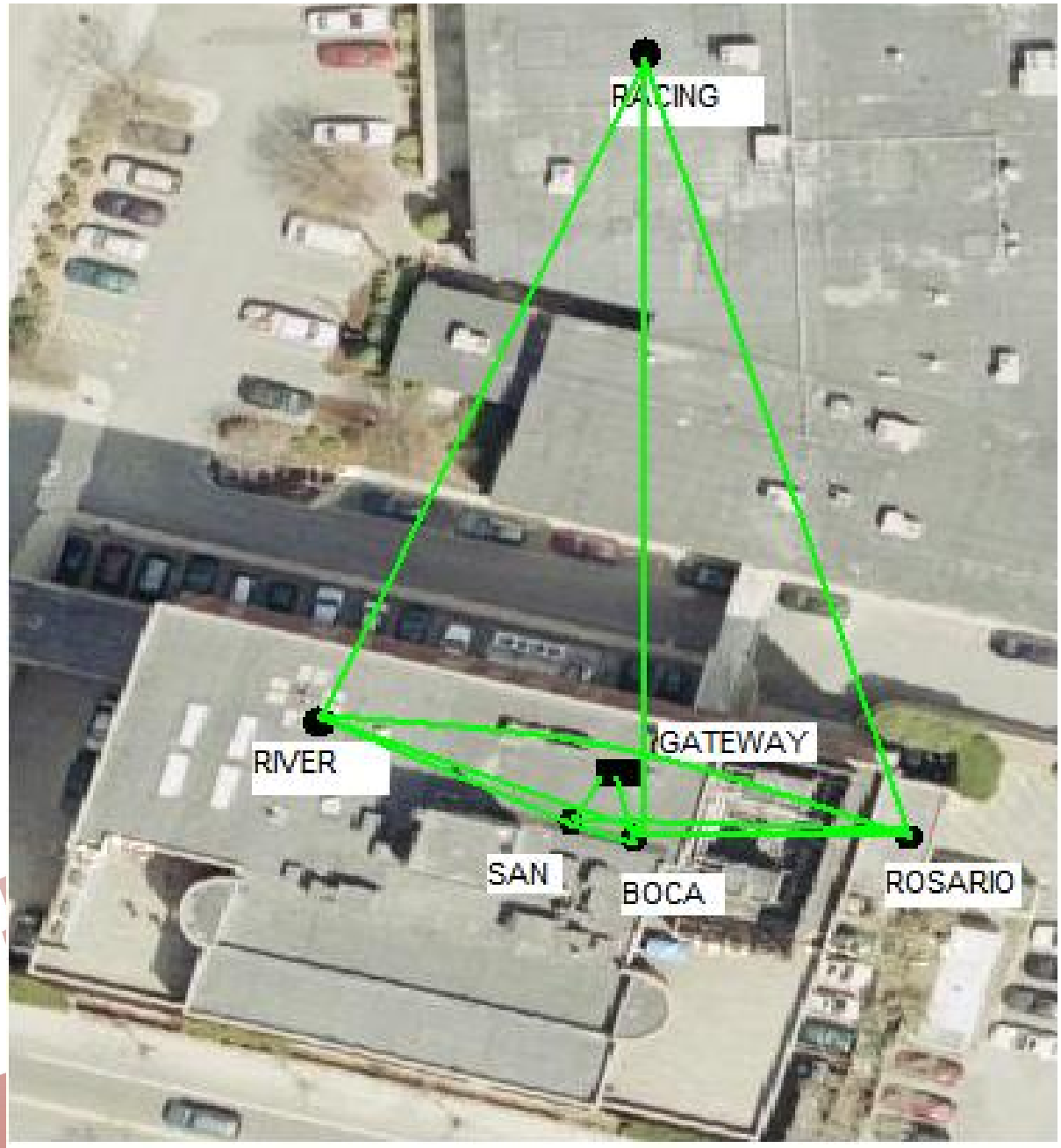
Early Node Prototype

Current Node Prototype



Harvard Rooftop Deployment

BBN Rooftop Deployment



Bidirectional Links

Sensor Nodes

Gateway Server

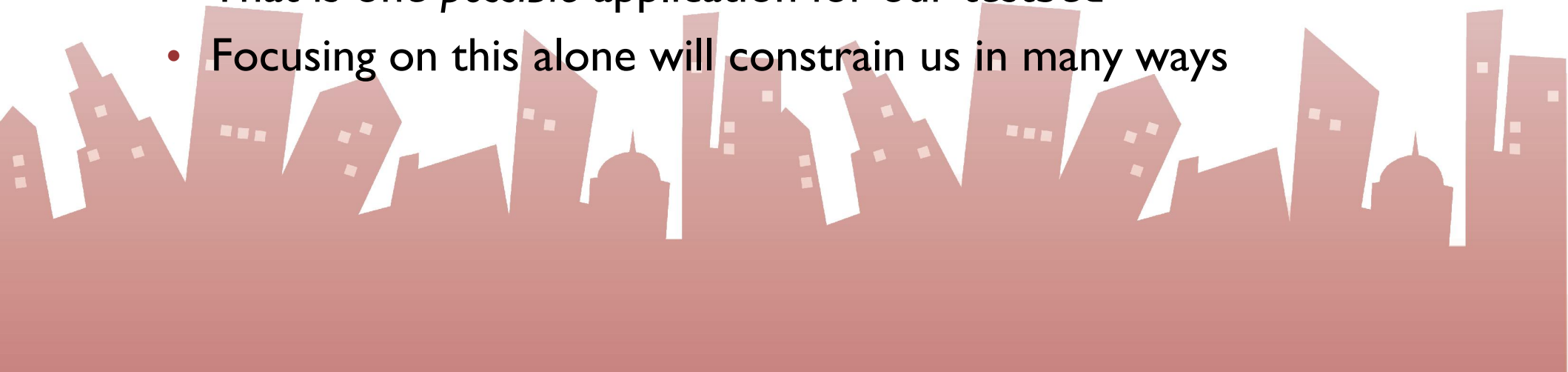
- 3 Indoor nodes plus gateway
- 2 nodes on roof of buildings
  - Racing
  - Rosario
- Fully connected except for Gateway

CitySense Networking

- Most nodes will use wireless mesh for connectivity
  - A few nodes will have wired connectivity and act as gateways
- Dual radio node design
  - "Backhaul" radio for management and monitoring mesh
  - "Experimental" radio for user applications
- Backhaul radio
  - Ubiquiti SR9 radio: 900 MHz, 802.11 b/g, 700 mW tx power
  - > 1 km range possible
  - Mesh configured using OLSR or other existing protocol
- Experimental radio
  - Wistron CM9 802.11a/b/g: 2.4/5.3 Ghz, 60 mW tx power

Some non-goals of this project...

- Reinvent mesh networking
  - Great deal of work has been invested in this; commercial solutions are out there.
  - Let's build upon the best work in this area and look at what can be done using mesh network as a foundation.
  - We will support *other* research groups wanting to experiment with new mesh routing protocols
- Provide public Wifi access
  - That is one *possible* application for our testbed
  - Focusing on this alone will constrain us in many ways



Vaisala weather sensor

Weather Transmitter WXT510

- Wind speed and direction
- Precipitation
- Barometric pressure
- Temperature
- Relative humidity

Well-calibrated sensor,robust packaging for outdoor use

- Designed for precise measurement of environmental conditions
- More accurate than typical component sensors used on motes

RS232 interface for configuration and data access

Other sensors

- Air quality: PM10 particle concentration
  - TSI SidePak personal aerosol monitor

- Noise pollution
  - Want to avoid use of microphones: Use specialized hardware just to measure overall dB level

- Gases: CO_2, NO, O_3 etc.
  - Vaisala CO_2 sensor
  - Siemens GasFET sensor array on a chip based on semiconducting metal oxides

Some Research Challenges

Resource management and sharing

- How to support many applications? Fairness? Security? Sandboxing?

Programming models and languages

- What is the right programming abstraction for a city-wide sensor network? ssh into every node? SQL? Something in between?

Robustness and administration

- How to administer and maintain a network of 100+ nodes over a wireless mesh, without physical access?

More Research Challenges

Security and privacy

- How to prevent script kiddies from taking over the network?
- Access to raw data may be limited by various policies

Interfacing to mobile and low-power sensors

- Tie-in with data collected by cell phones, mote networks, vehicular sensors

Extensibility and customization

- Tension between supporting many users and customizing for specific application needs

Application Programming Model

- What is the right programming model for CitySense applications?
- Option #1: Bare Linux plus ssh
  - Very inefficient, every user must built up own monitoring, management, and directory services
- Option #2: pwn the node...
  - One user at a time, flash new custom OS+app environment for each user
- Option #3: Managed application environment
  - Common service responsible for deploying, starting, stopping, and decommissioning applications
  - Scheduling dictated by global management policies
-

Resource Management

- As an open testbed, CitySense will be subject to widely varying demands from different users.
  - How can we provide these users with "good enough" service while still permitting some degree of timesharing?
- Node resources are somewhat scarce:
  - Slow CPU, limited RAM, poor network connectivity, no disk
- Best-effort timesharing unlikely to work
  - Not enough resources to just let users "go at it"
- Should we do batch scheduling?
  - Appropriate for short-running experiments requiring high fidelity
  - But shuts out users doing lightweight data-gathering

Remote Maintenance

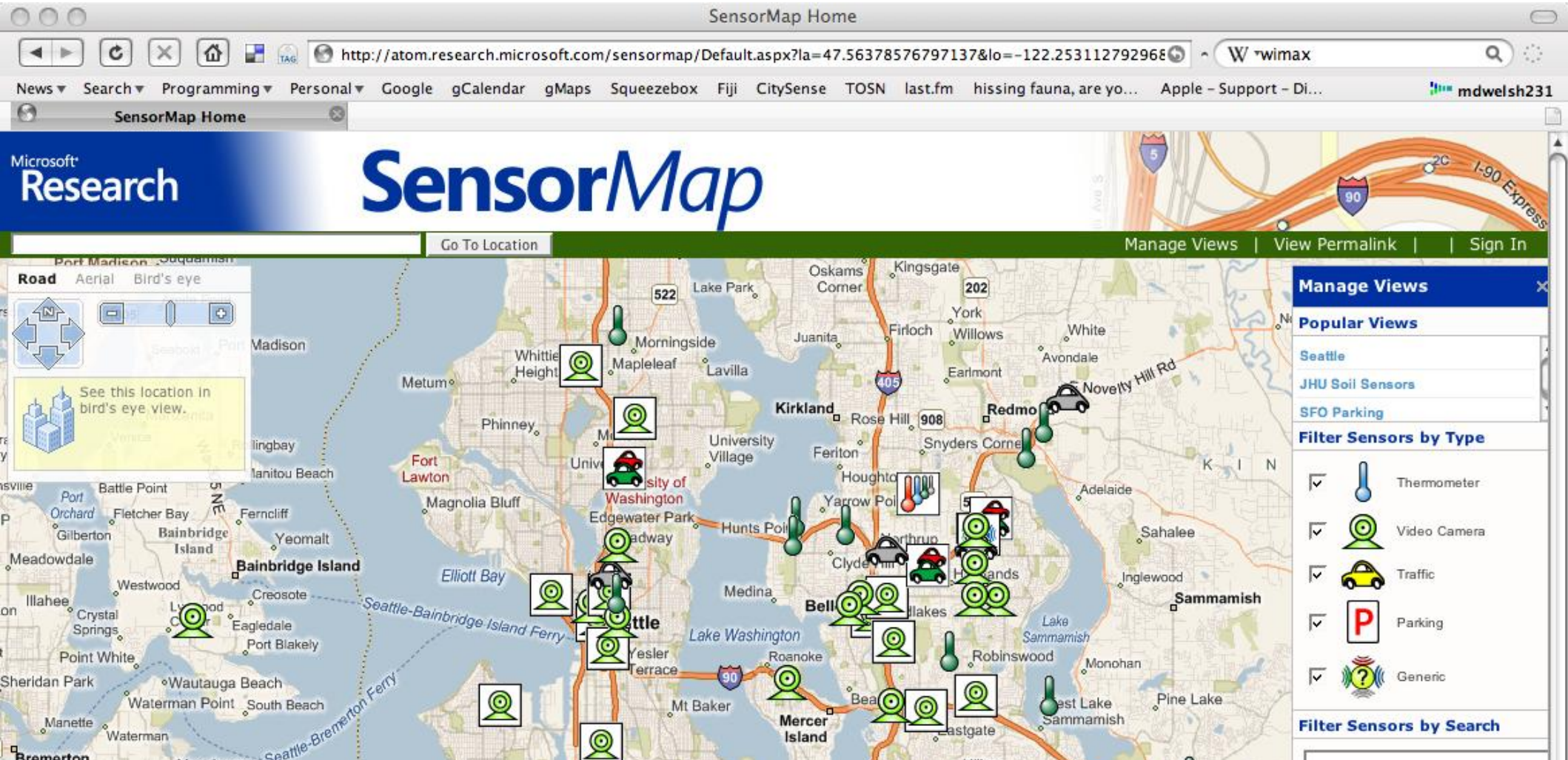
- We will not have physical access to nodes once deployed!
  - Need to ensure that nodes can be recovered to "known good" state in case of software crash or hang
- Soekris supports on-board hardware watchdog
  - Daemon checks node health and clears watchdog register only if things seem to be OK
  - Force reboot if network connectivity lost, memory leak, etc.
- Grenade timer
  - Use cheap lamp timer to do hard reset once a day
  - (Nearly) foolproof and no need for fancy software support

Software updates

- Must periodically upgrade base node OS and frequently push out new application code
  - Doing this in a naïve way, i.e., scp to each node, will perform very poorly on the wireless mesh
- Possible approaches:
  - Network coding (Vlah and Kung approach), Bittorrent, etc.
- Must work correctly in the face of node failures and network connectivity loss
  - Cannot assume all nodes are up all the time
  - Our approach: rsync over spanning tree
  - Permits asynchronous updates; can extend to leverage multipath

Data Visualization

- Microsoft SensorMap infrastructure
  - http://atom.research.microsoft.com/sensormap/
  - Shows real-time sensor data; performs spatial aggregation



Node monitoring

- Currently using Ganglia to monitor node status
  - Daemon runs on each node and gathers statistics on CPU, memory, bandwidth usage, number of processes, etc.
  - Data reported to central server for storage and visualization
  - Uses 'rrdtool' rather than persisent storage of raw node reports
  - Multicast UDP support for sending status: May work well on a wireless mesh

Ganglia:: Cluster Report

http://www.citysense.net/ganglia/?m=load_one&r=week&s=descending&c=CitySense+Server&h=&sh=1&hc=4

wimax

News ▾ Search ▾ Programming ▾ Personal ▾ Google gCalendar gMaps Squeezebox Fiji CitySense TOSN last.fm hissing fauna, are yo... Apple – Support – Di... mdwelsh231

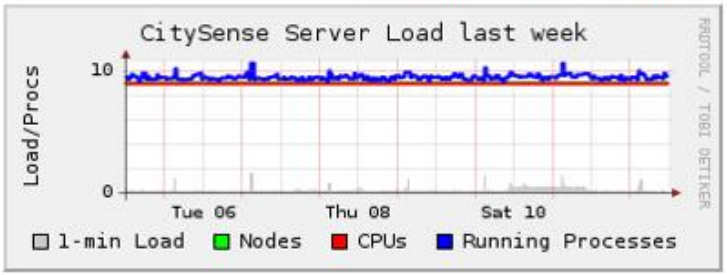
Ganglia:: Cluster Report

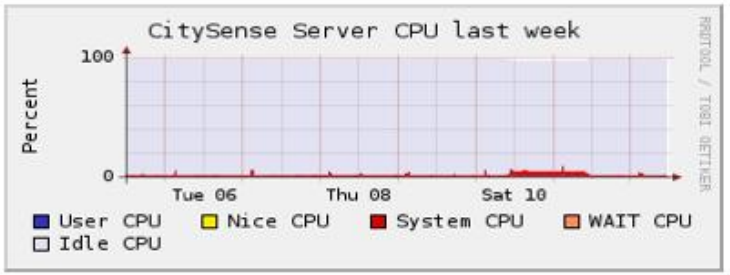
Overview of CitySense Server

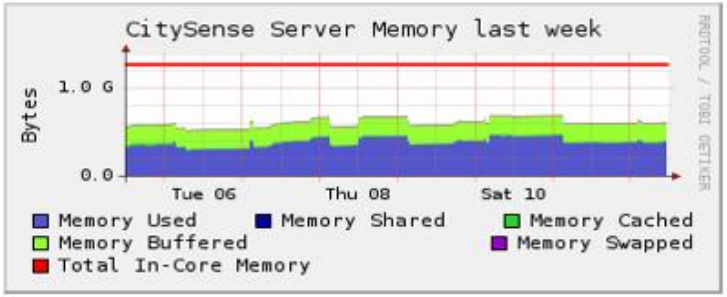
CPUs Total: 9
Hosts up: 9
Hosts down: 0

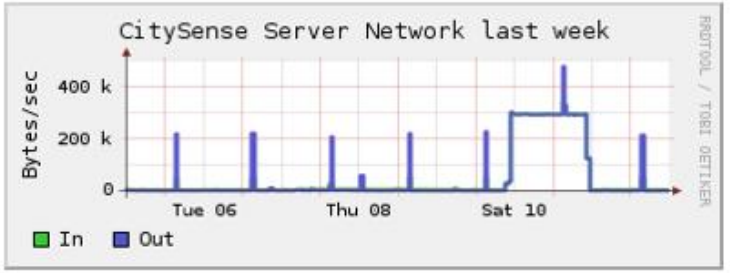
Avg Load (15, 5, 1m):
0%, 0%, 0%
Localtime:
2007-11-12 11:46

Cluster Load Percentages

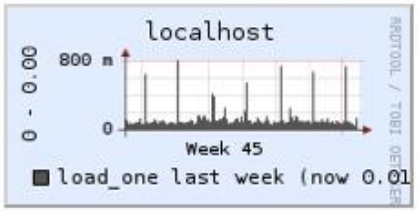


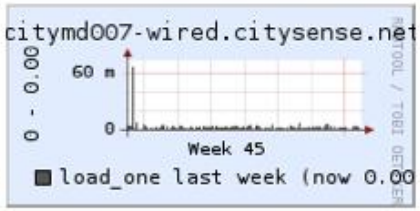


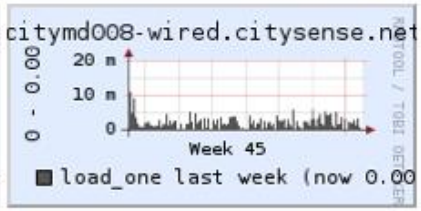


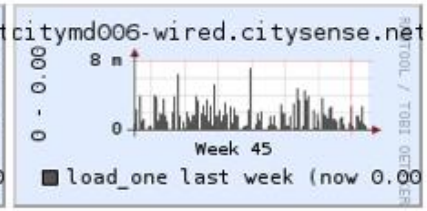


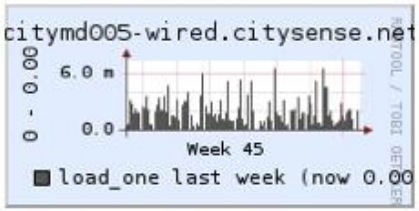
Show Hosts: yes no | CitySense Server load_one last week sorted descending | Columns 4

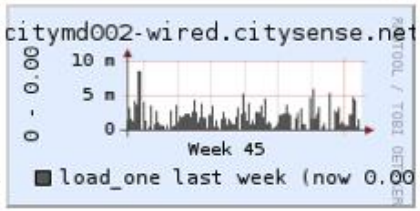


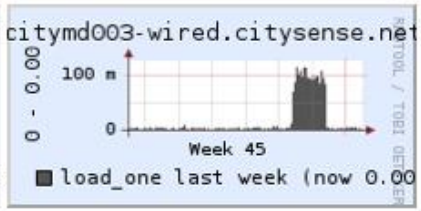


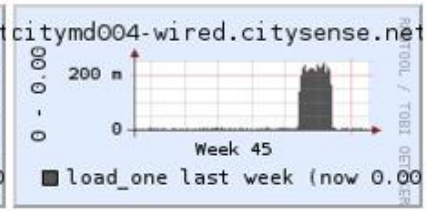


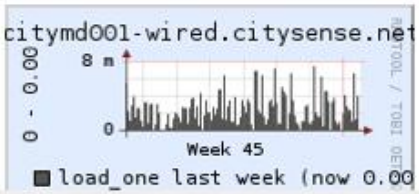












Done

2.360s

Collaborators

Majid Ezzati: Co-PI Harvard School of Public Health → Urban pollution studies

Ken Mandl: Director of CHIP's program Childrens Hospital, Boston → real-time tracking of ER symptom reports

David Gute: Tufts University EE department: water quality sensors

Tom Little: BU EECS: video sensors

Chris Rogers & Marina Bers: Tufts EE: Educational Outreach → K-12 curriculum in sensor nets.

Summary

- CitySense presents huge opportunity for the sensor network community
  - Develop, deploy, and experiment with sensor networks at scale in complex real-world outdoor urban environment
  - Shared research facilities for supporting diverse research groups
- Planned 100-node outdoor testbed in Cambridge
  - Linux-based embedded PCs with 802.11 and professional weather sensor
  - Planned future sensors include pollution/smog sensors.
-

Thanks

CitySense is supported by grants from the National Science Foundation and Microsoft Corporation.

Contact: mdw@eecs.harvard.edu

http://www.citysense.net