Lab Plans

- Saturday: 2:00
  - Milestone 6 due this day
- Next week: (by default) same schedule
  - You can also arrange for one of us to let you in

Administrivia

- Milestone 6 due Saturday
- Widget demos due Friday the 19th

Exam Preparation

- 2 hour exam
  - Monday December 15th, 10:30
  - LGRC 201
- Closed book, closed notes, closed electronic device (mechanical computing devices allowed)
- Look to the homework assignments and quizzes for example problems
- Please post suggested problems on the news group
  - Those posted by Saturday may be on the exam

Basic Physics

- Electrons (free versus not)
- Positively-charged atoms (and holes)
- Conductors versus insulators

Digital Electronics

- Representing “0” and “1”
- Driving a line versus letting the line float
- Debounce
- Keypad scanning
- Encoders

Electronics

- Kirchoff's current and voltage laws
- Ohm's law and resistance
- Current across a capacitor
- Voltage dividers
- RC-circuits and time constants
- Derivation of series and parallel resistance/capacitance
- Know your units (Farads, Ohms, Voltage, etc...)

Diodes

- Restrict current flow
- Forward voltage
- LEDs

When two LEDs of different color are placed in parallel in the circuit, why does the red one emit more photons?

Semiconductors

- N- and P- semiconductors
- Depletion regions
- Building a diode
- Building a transistor
  - Threshold: forward voltage on the base-emitter diode
  - Current amplification
  - NPN versus PNP transistors
- Implementing basic logic gates with NPN transistors

Op-Amps

- Basic functionality
- Ideal properties
- Application in voltage amplification
  - Positive gain circuit versus negative gain circuits

Synchronous Serial Protocols

- Master/slave architectures
- SPI versus I2C
  - Physical connection
  - Addressing and control
  - Clocking
  - Data flow
    - Unidirectional versus bidirectional
    - Daisy chaining (in SPI)

Asynchronous Serial Protocols

- Clocks and timing
- Error checking
- Flow control

PIC Architecture

- Memory architecture: what is the basic form and why?
- Timer basics (any timer questions will be accompanied with the corresponding timer diagram from the PIC specification)
- Interrupts and interrupt handling
- I/O pin configuration (in the abstract) and current driving limits

Software Abstractions

- Critical sections
- Very basic real-time scheduling

HW/Quiz Questions?

Now is the time....

Wearable Computers as Packet Transport Mechanisms in Highly Partitioned Ad-hoc Networks

J. A. Davis, Andrew Fagg, Brian Levine

Department of Computer Science
University of Massachusetts, Amherst

Pedestrian Network (PEDnet)

- Increasingly small, inexpensive sensors and data storage devices with local, low-power communication
- Wearable computers, palm tops, cell phones with a variety of communication capabilities
- How do we build a network?
- Mobile devices become the packet transport mechanism
- Must deal with limited buffer and computational resources

An Inspirational Packet Router

Applications

- Widely distributed sensing: data gathering
- Smart office spaces: conference rooms with schedule beacons, cabinets with content beacons
- Devices that can "call" for support: plants for water, printer for paper, vending machine for refills

Highly Partitioned Ad-hoc Networks

- Ad-hoc network--a collection of cooperating nodes communicating without a pre-deployed infrastructure
- Remove constraint that a path exist between sender and receiver--now it's partitioned
- New routing protocols are needed for partitioned networks

Approach

- A number of mobile and stationary agents generating messages to random destinations
- *Epidemic Routing* [Vahdat, Becker]: Two devices meet: exchange messages that are not held by both--buffers fill quickly
- Buffer full: drop packets according to some *drop strategy*
- Heuristic strategy: Drop oldest packet
- Our strategy: Drop the packet which is least likely to be delivered

Estimation of Meeting Likelihood

- Temporal Difference (TD) method for estimation.

$$\boldsymbol{M}_{t+1}(A,C)=\begin{cases}\lambda M_t(A,C) & \text{if none are co-located}\\ \lambda M_t(A,C)+1 & \text{if } C=B\\ \lambda M_t(A,C)+\beta M_t(B,C) & \text{if } C\neq B\end{cases}$$

$\boldsymbol{M}_t(X,Y)$ is the estimated likelihood of X meeting Y

A the agent performing the update

B a co-located agent

C all agents

Drop Least Encountered Strategy

- When A meets B, it orders the messages according to:

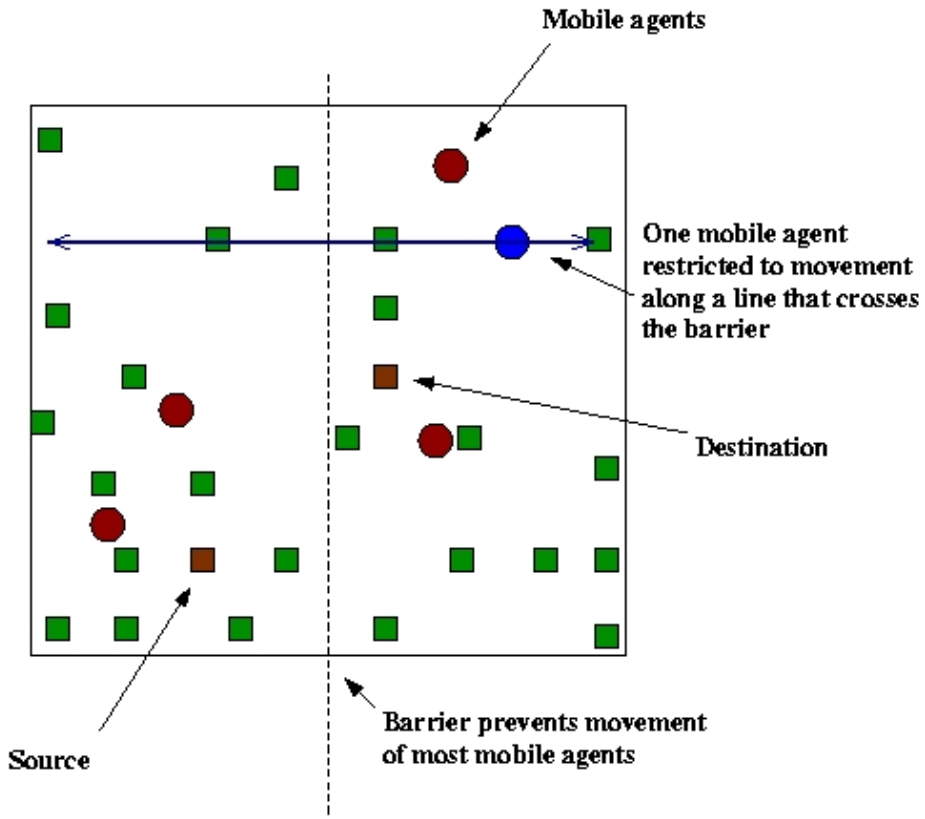
$$\boldsymbol{M}_t(A,C) - \boldsymbol{M}_t(B,C)$$

- **A** keeps the top *N* messages (and likewise for **B**)

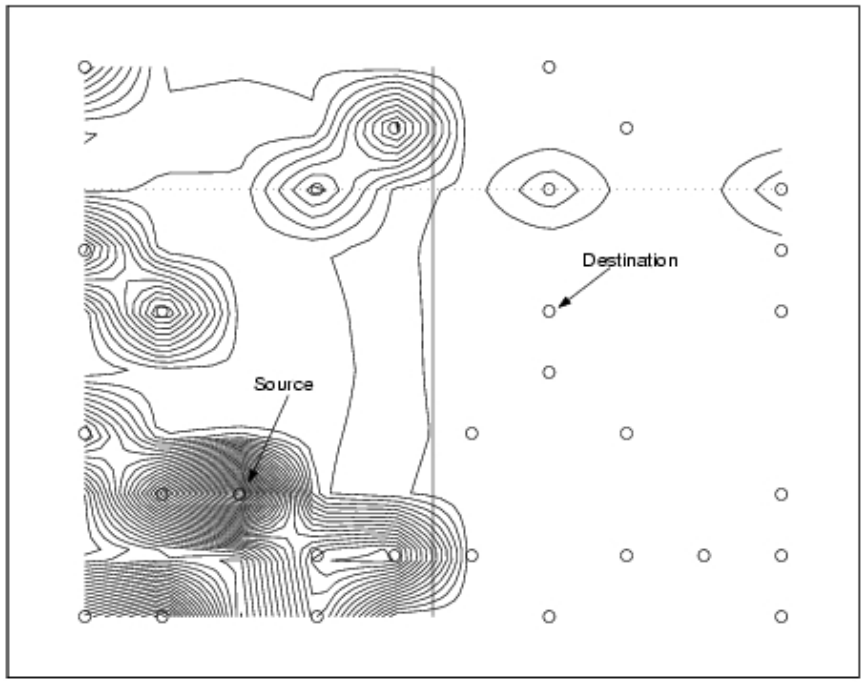
- Side effect: no messages are dropped.

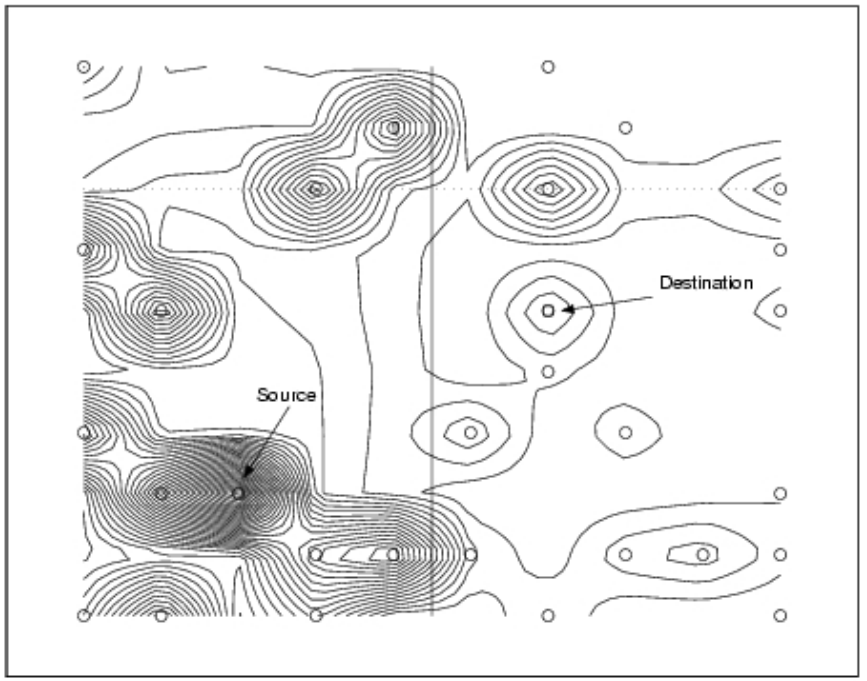
Experimental Environment

- Unstructured: Mobile agents free to move
- Restricted: Mobile agents restricted to random movements within one half of the environment with one agent moving deterministically across boundary



Packet Exposure





- Drop Oldest
- Drop Least Encountered

Summary

- Estimation of meeting likelihood is an inexpensive, local operation (via TD)
- Possible to incorporate the cost of message exchange in the decision process
- Will soon include dynamic message time-outs and message acknowledgments
- Working to include mobile nodes equipped with 802.11 -> mobile wormholes

Querying the Physical World

Challenge:

- Large number of sensors distributed over a wide area
- Limited bandwidth and spurious network connectivity

- How do we support sensor queries in some uniform way?
  - Supporting historical queries
  - While minimizing query delays due to network issues

One Approach

Treat the system as a large database

- All sensors deliver their data to a central database (a warehouse)
- Becomes a standard database query problem

- ... but does not address network bandwidth and delay issues

The Proposed Solution

- Cache sensory data on a set of machines that are constantly connected to the network (and that have access to bandwidth)
- Each of these machines collects data from a subset of the available sensors
- By caching, we solve the problem of network delays to individual sensors

Changes to the Query Language

- Could still perform a query at the central site, but this means we potentially would have to move a lot of data.
- Solution: distribute queries (or parts of queries) to the caches
  - The central site then receives the results of these queries

Changes to the Query Language

Can also push parts of queries out to the sensors:

- Report an event
- Report data values whenever certain conditions hold
- Report data value at regular intervals

- One critical question: how much of the decision making do we push out to the sensor?

Changes to the Query Language

- The database domain already has standard techniques for reasoning about how a query should be executed
- We add some new measures of query performance, e.g.:
  - Time required to execute the query (actually, this is a pretty standard measure already)
  - The amount of energy required to execute the query and transmit the necessary data

Questions/Extensions

- Would like to have a way to express the tradeoff between fresh data and query time
- Since queries can be quite extended in time: several queries may compete with one-another for computation and bandwidth resources
  - Sensors and data collection nodes must be smart enough to deal with conflicting requests
  - Would like to have ways to resolve competition (possibly through some priority scheme)

Questions/Extensions

Would like to think about mobile nodes:

- How to structure queries if they are reporting to different collection sites?
- Would like to have node location as an available resource in the query planning process
  - Query planning drives control of the mobile nodes!
  - Could also include bandwidth constraints in the control process