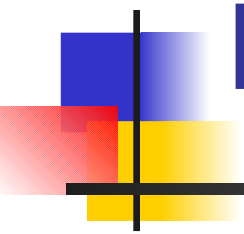
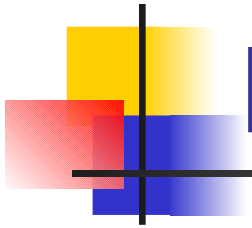


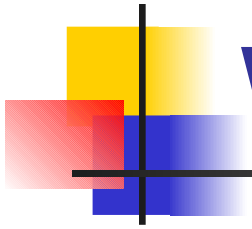
Geographic Routing Without Location Information

AP, Sylvia, Ion, Scott and Christos



Routing in Wireless Networks

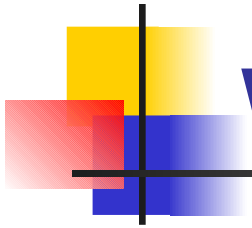
- Distance vector
  - DSDV
- On-demand
  - DSR, TORA, AODV
  - Discovers and caches routes on demand
- Geographic
  - GPSR - scales very well



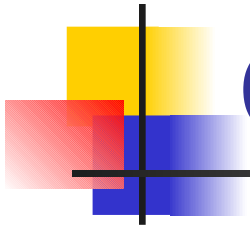
What is the problem?

- No address aggregation
  - Except geographic routing, all other approaches require O(N) state per node
- Routing by coordinates is a good way to avoid O(N) per-node routing state

Why geographic routing without location information

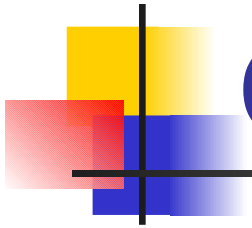


- GPS takes power, doesn't work indoors
- Obstacles, non-ideal radios
- Coordinates computed will reflect true connectivity and not the geographic locations of the nodes



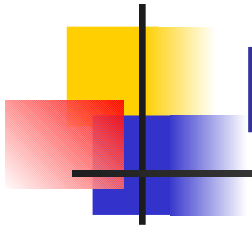
Geographic routing

- Choose coordinates for nodes
- Greedy routing
  - Proceed closer to destination at each hop
- How to deal with voids?
- "Addresses" of nodes keep changing as they move
  - Need a lookup service for the current location of a node
  - Can be done using a DHT (as in DCS or GLS)



Outline

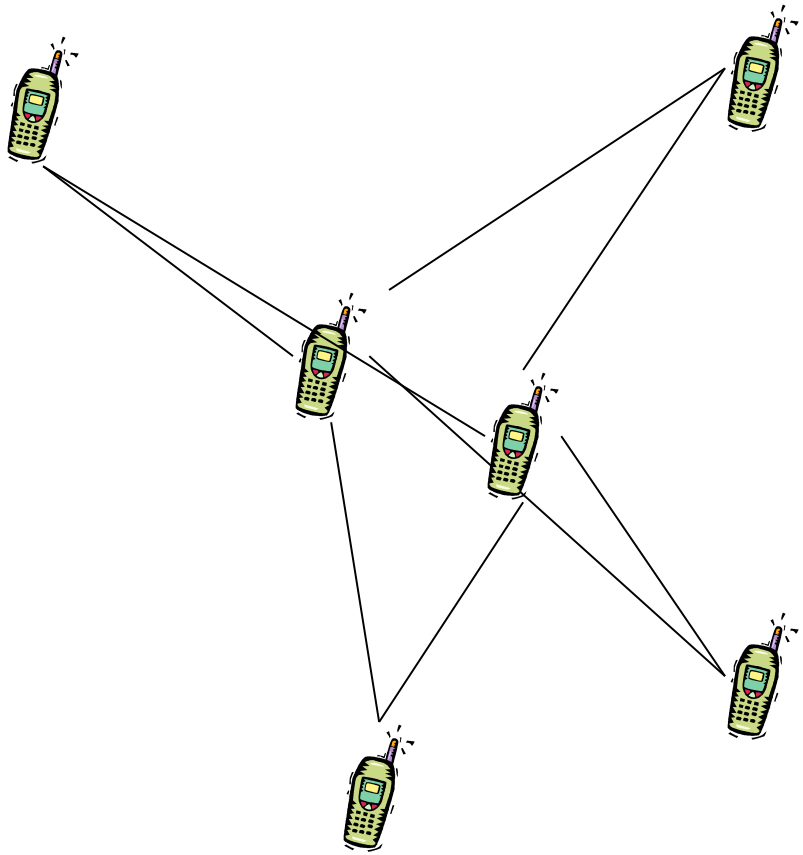
- Perimeter nodes and their locations are known
- Perimeter nodes are known but their locations are not known
- Nothing is known about the perimeter
- Dealing with Mobility



Rubber Bands

- Iterative process for picking coordinates for a node
- Some nodes along the periphery of the network know their correct (relative) locations and are fixed
- Other nodes compute coordinates by relaxation
  - Assume that nodes are connected by rubber bands and slowly converge to the equilibrium

Rubber Bands



Every node moves to the average of its neighbors coordinates at each step in the iteration

Original Shape of the Simulated Network

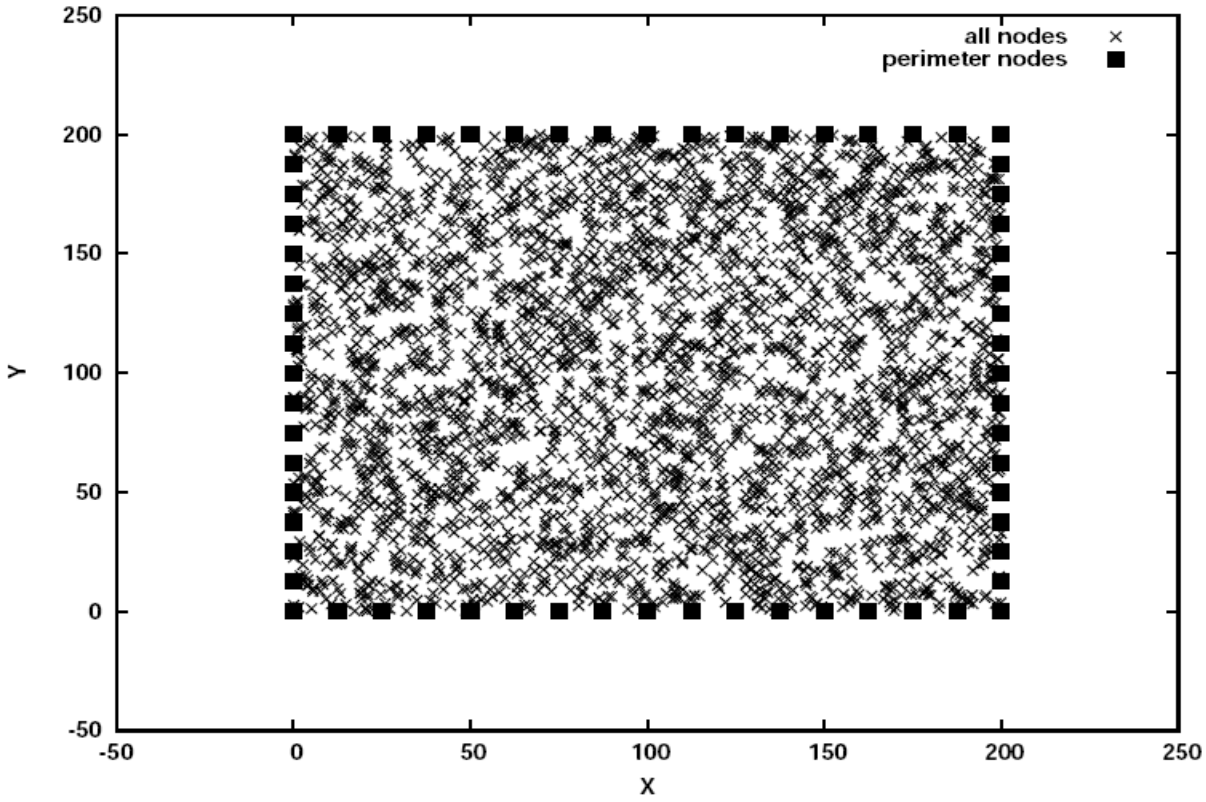
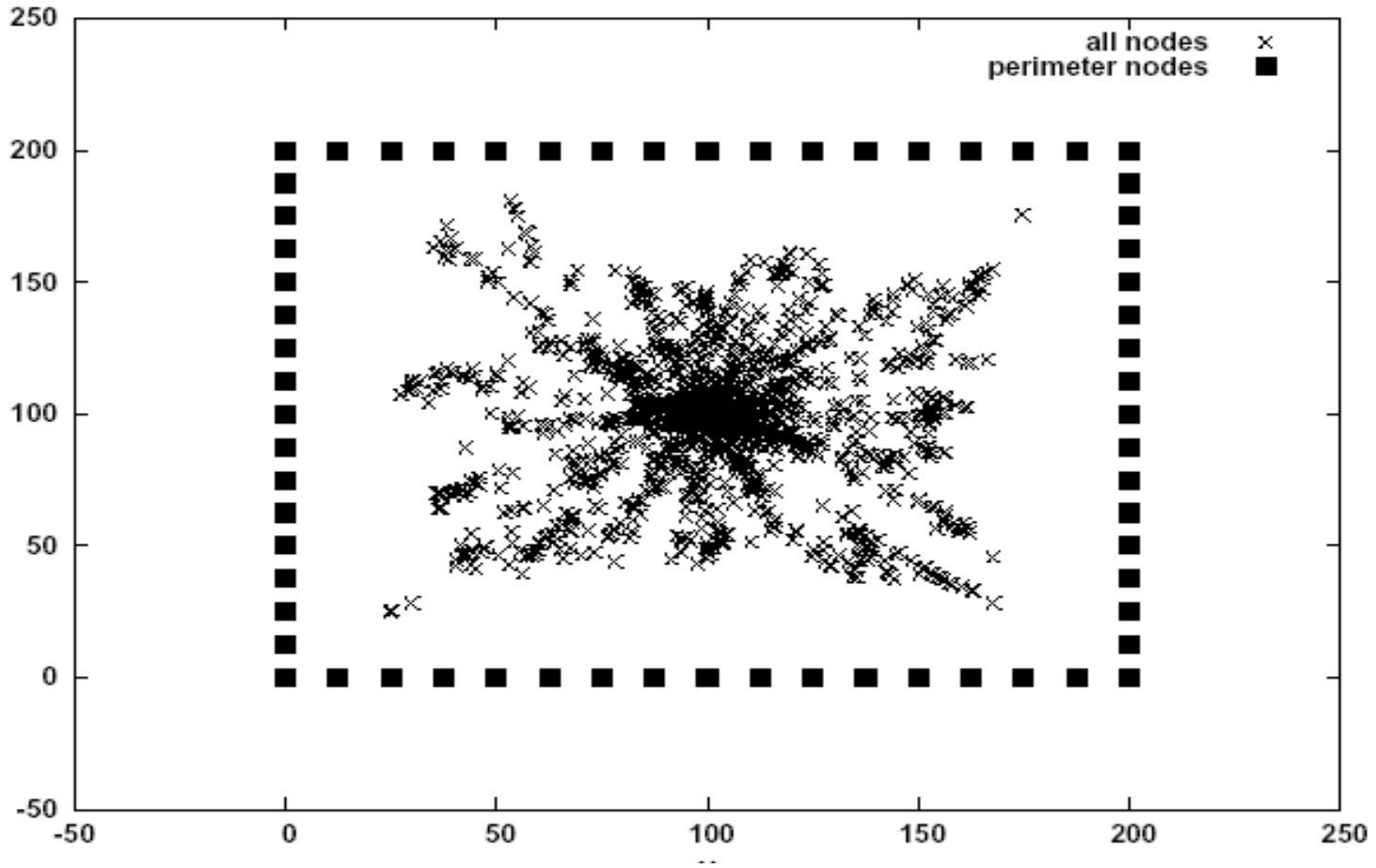
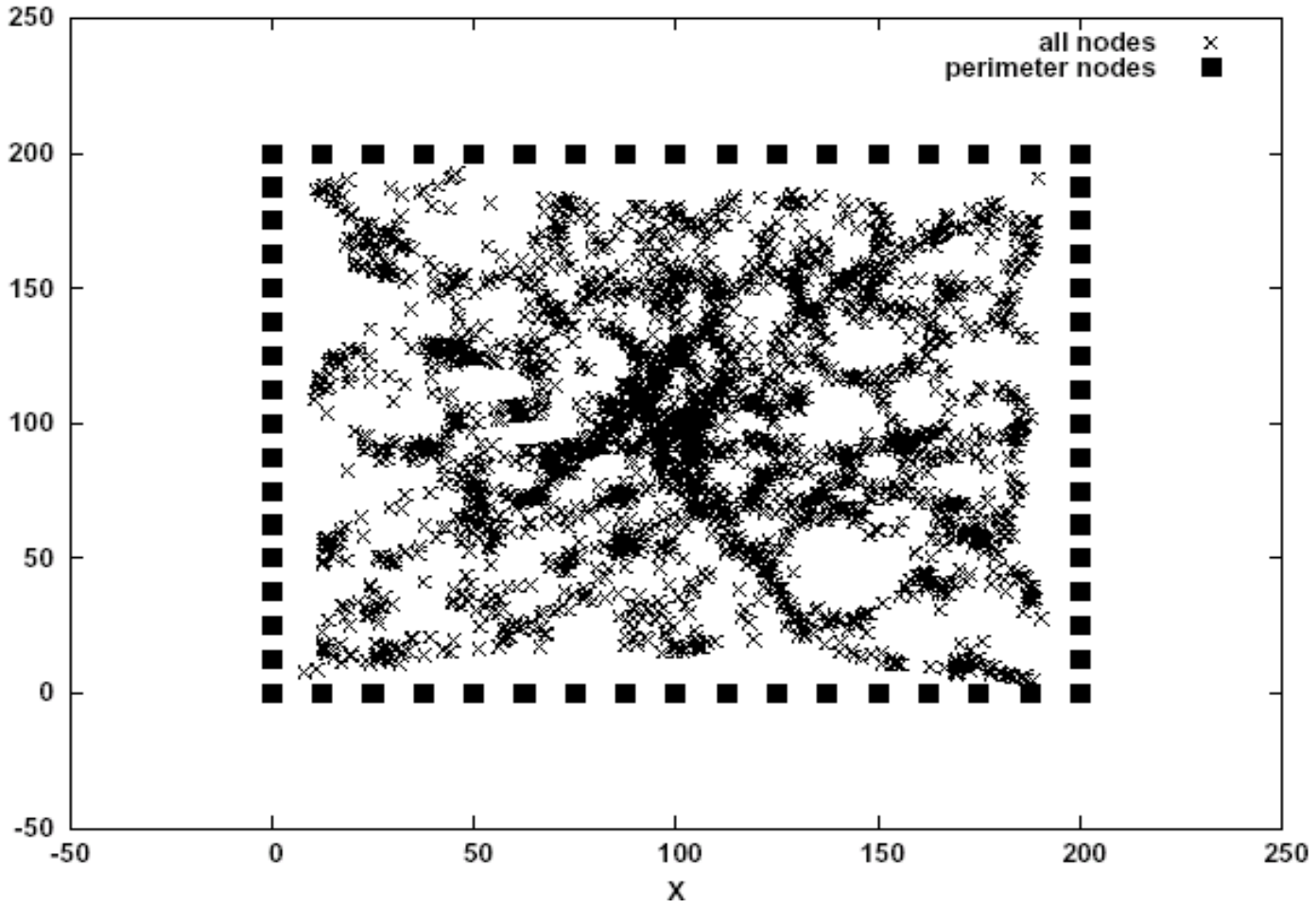


Figure 1: A network with 3200 nodes. Perimeter nodes are represented by black squares. The radio range of each node is 8 units.

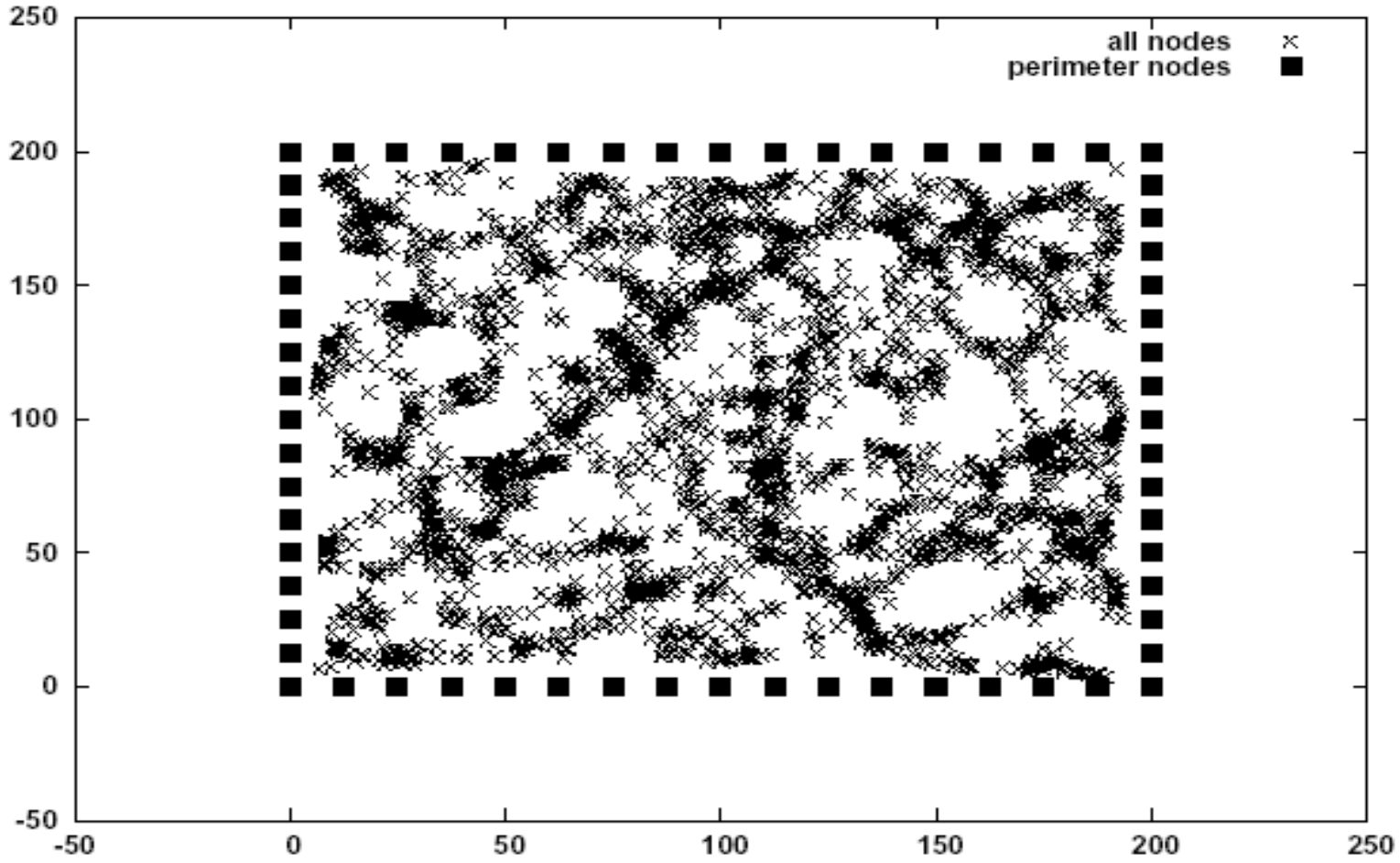
Perimeter nodes are known (10 iterations)



Perimeter nodes are known (100 iterations)



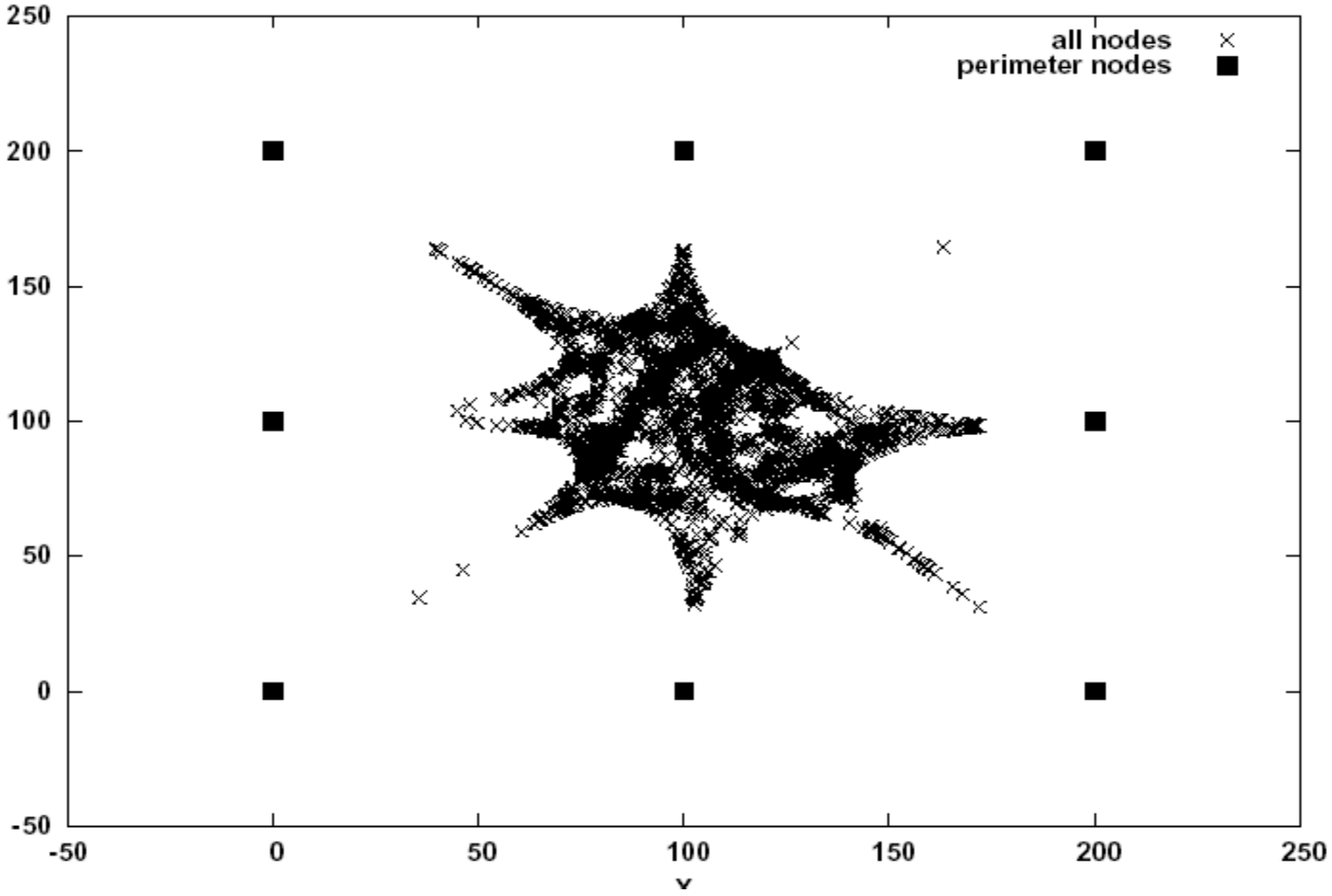
Perimeter nodes are known (1000 iterations)



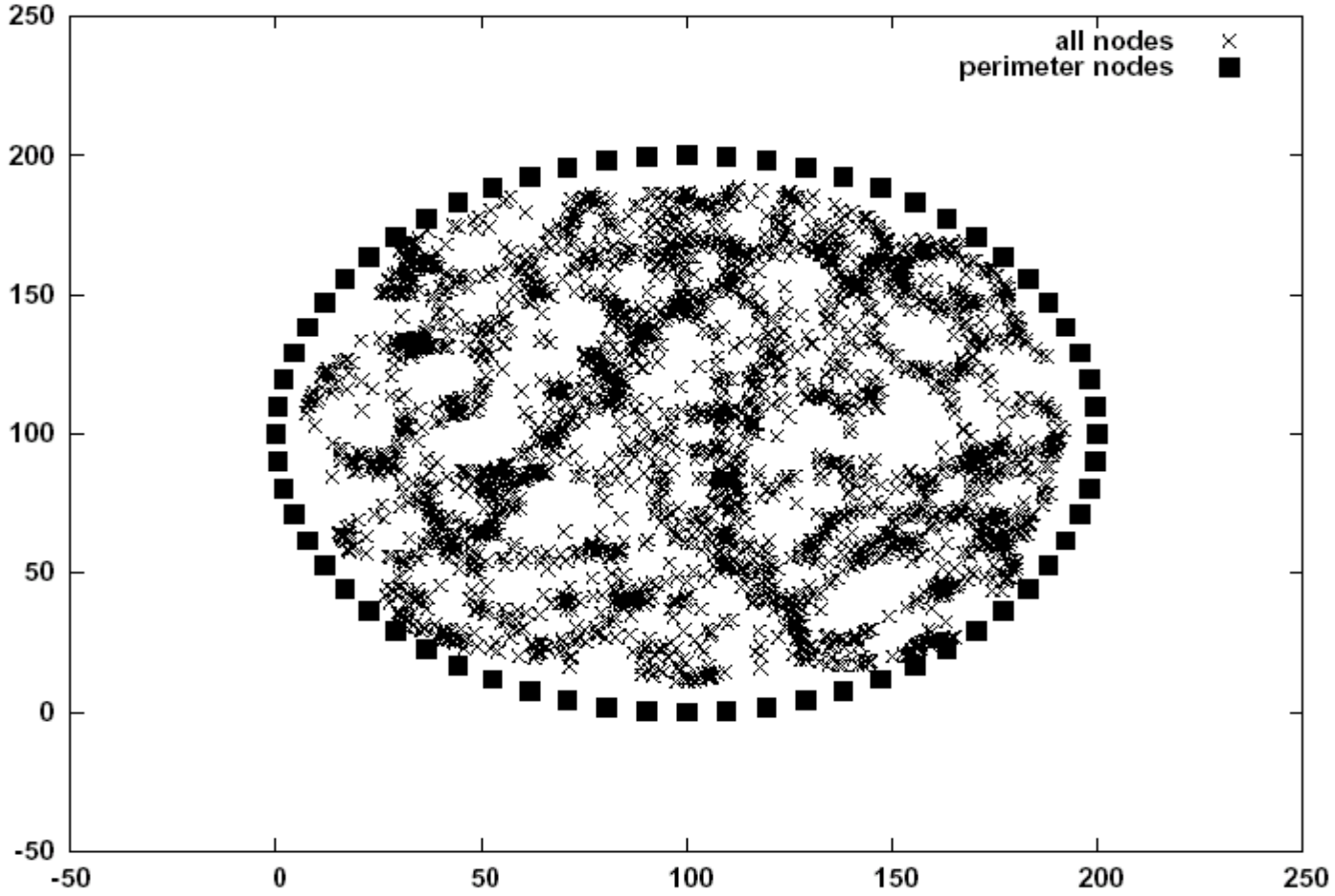
Rubber Bands (implementation and overhead)

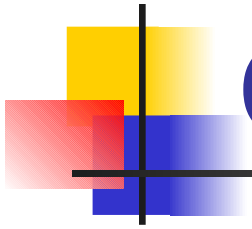
- We need a periodic heartbeat between neighbors so each node can maintain a list of its neighbors
- We just send the current position of the node along with the heartbeat packet it broadcasts
- Each time a heartbeat packet is received, we recompute the coordinate

Resiliency of the rubber band approach - I



Resiliency of the rubber band approach - II

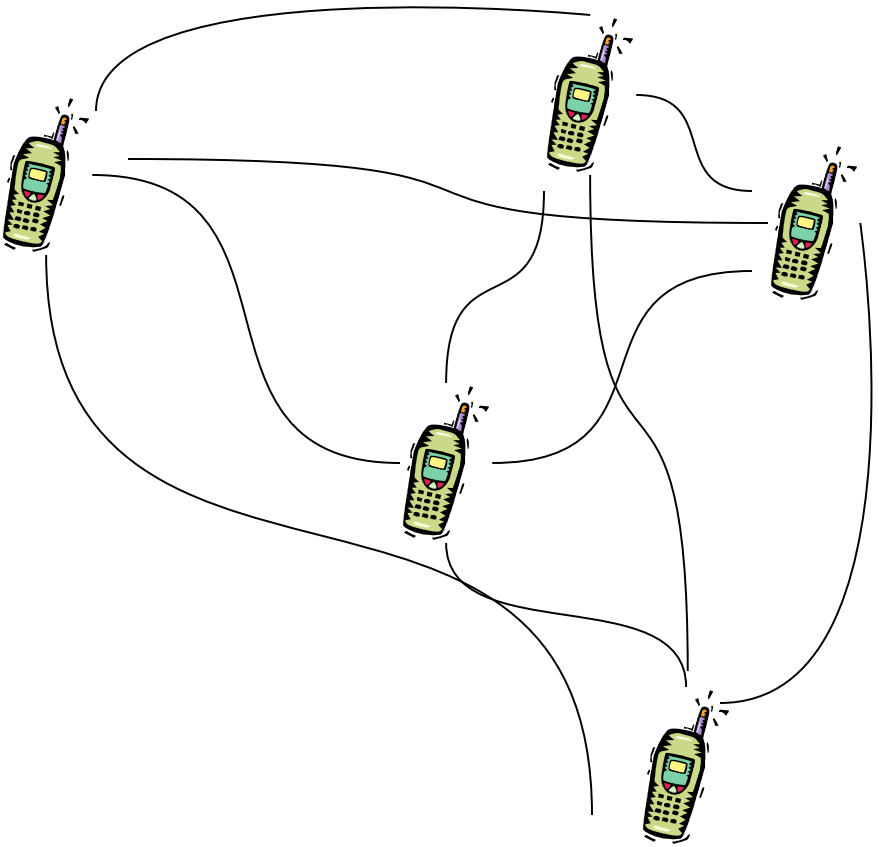




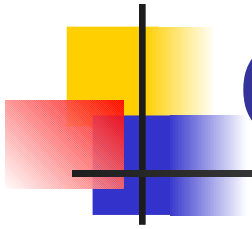
Outline

- Perimeter nodes and their locations are known
- Perimeter nodes are known but their locations are not known
- Nothing is known about the perimeter
- Dealing with Mobility

Balls and Springs



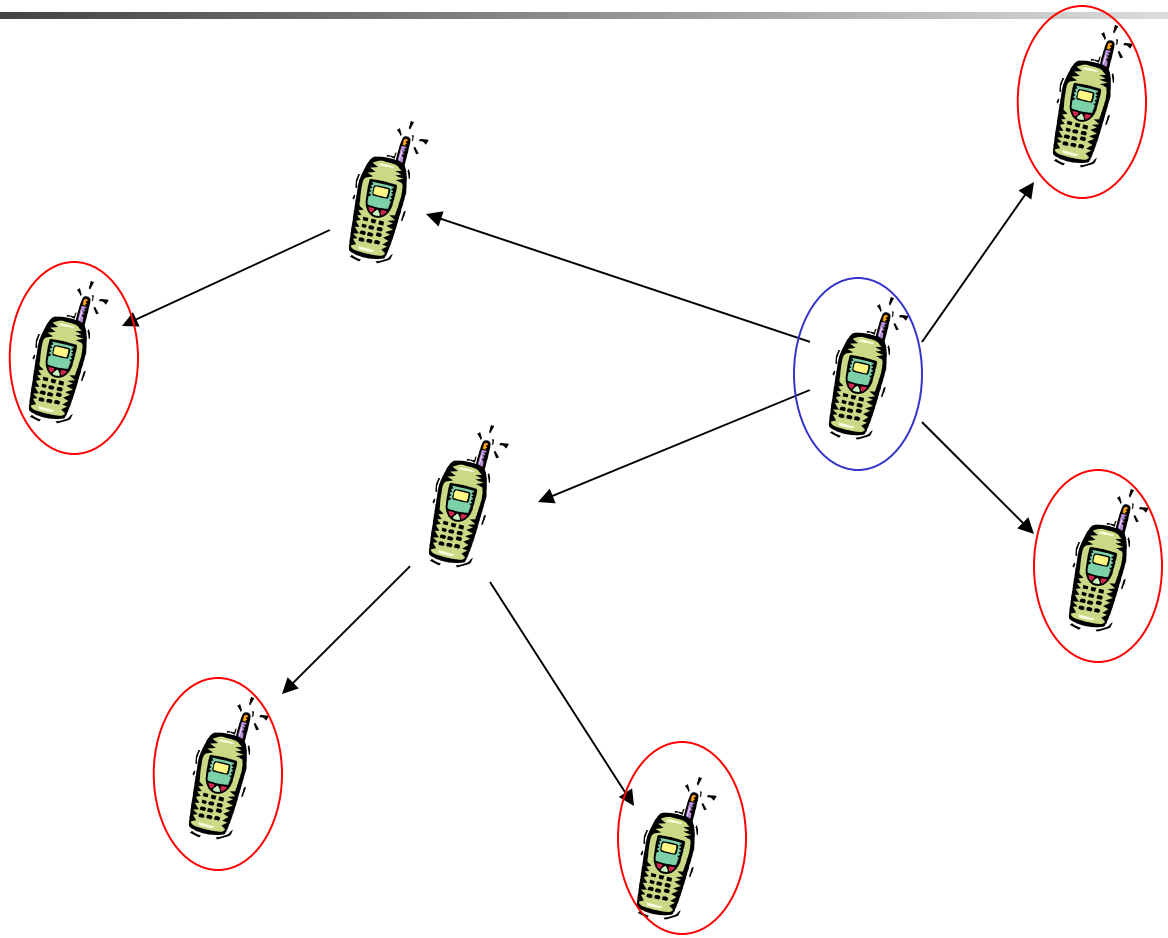
- A useful technique to get fairly accurate positions for a bunch of beacons given the all the inter-beacon distances (in number of hops)
- Assume that they are connected by springs of length proportional to the number of hops

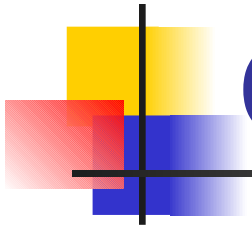


Outline

- Perimeter nodes and their locations are known
- Perimeter nodes are known but their locations are not known
- Nothing is known about the perimeter
- Dealing with Mobility

Perimeter node detection

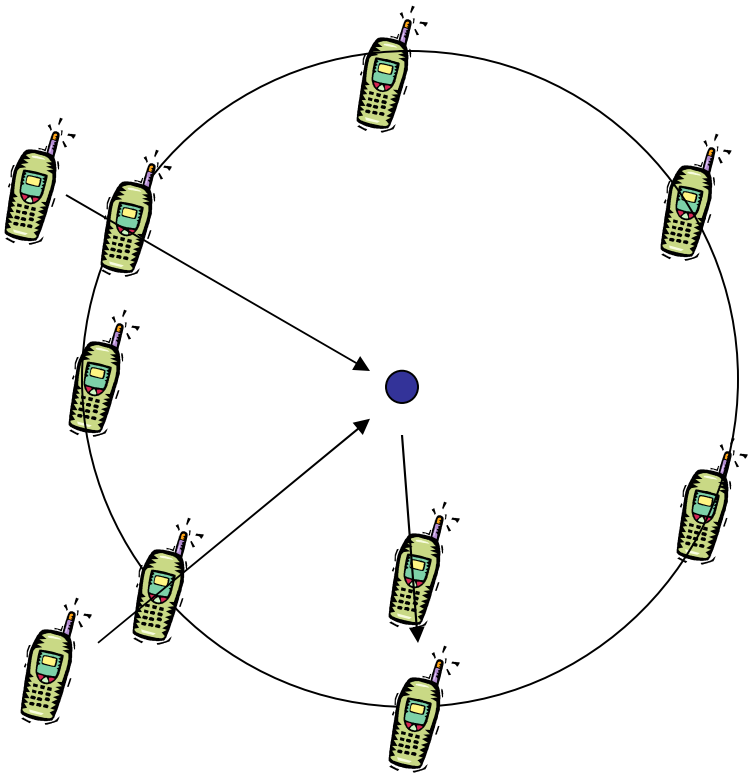




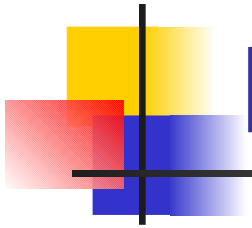
Outline

- Perimeter nodes and their locations are known
- Perimeter nodes are known but their locations are not known
- Nothing is known about the perimeter
- Dealing with Mobility

Perimeter nodes on circle



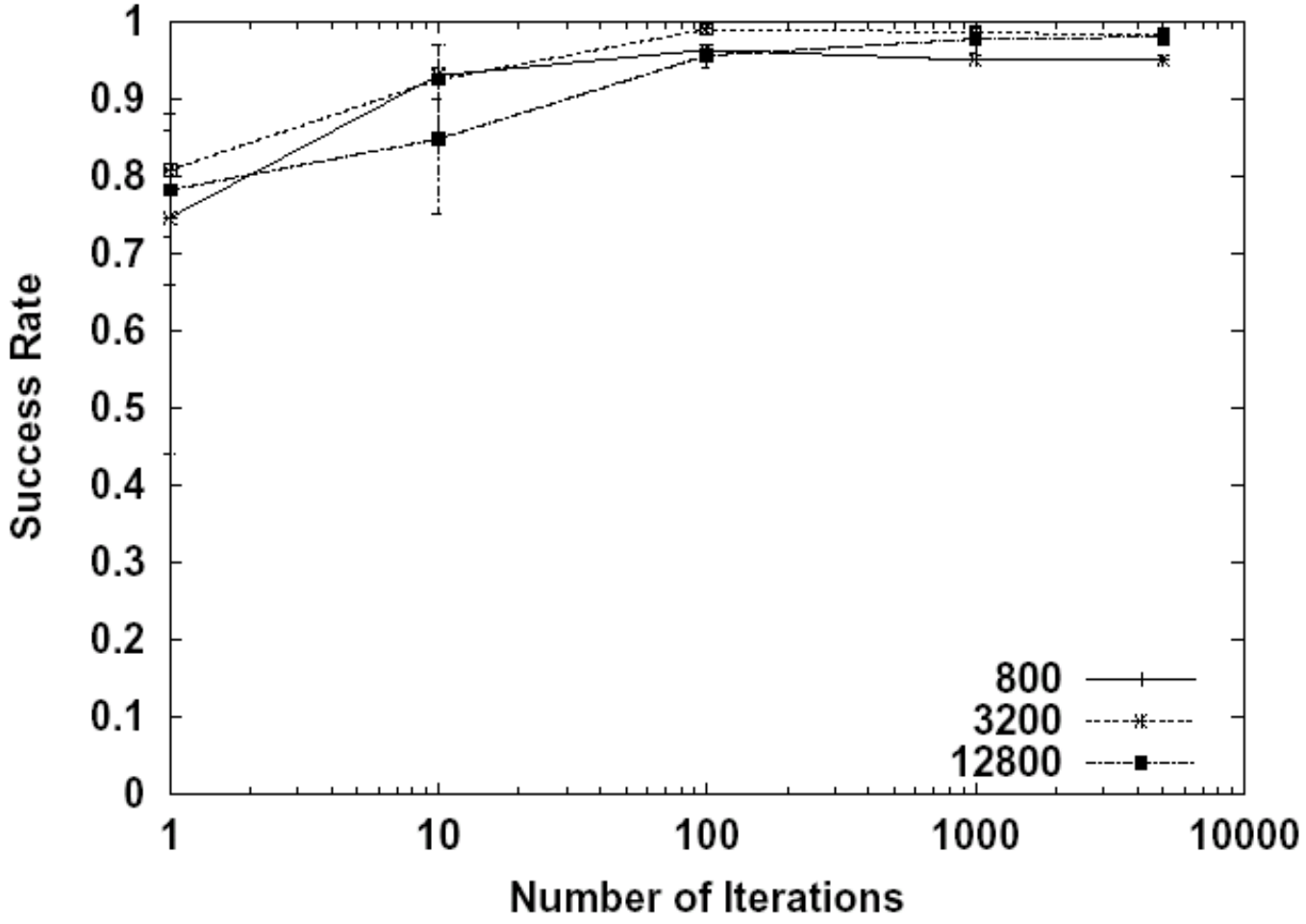
- Prevents continual shrinkage of the virtual geometry
- Make it easier to implement a DHT
- Steady state overhead is independent of the size of the network



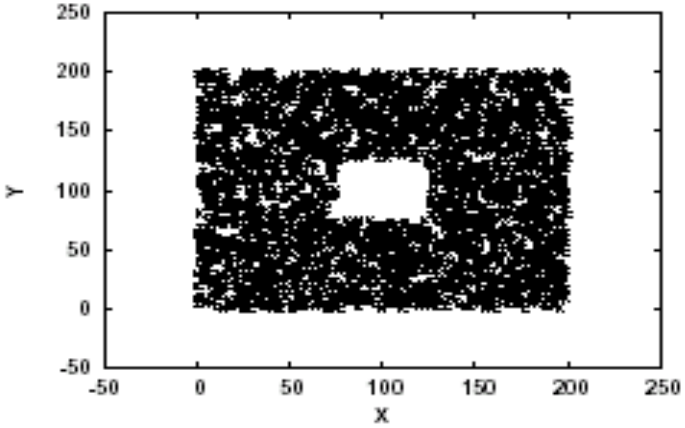
Results

- Event driven packet level simulator
- Doesn’t model application traffic or collisions
- Scales to 3200 nodes with packet events and 128000 nodes without events
- 3200 nodes distributed randomly in a 200x200 square. Radio range is 8, density is held constant while scaling up

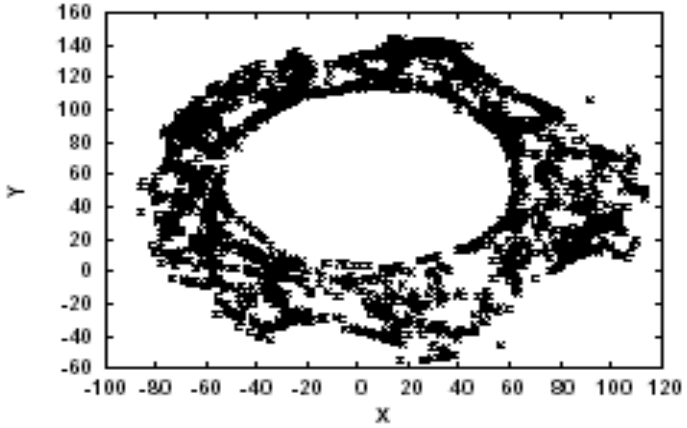
Success rate of greedy routing



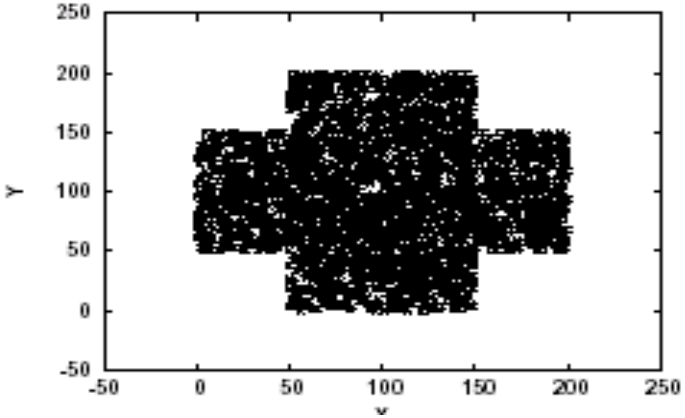
Weird Shapes

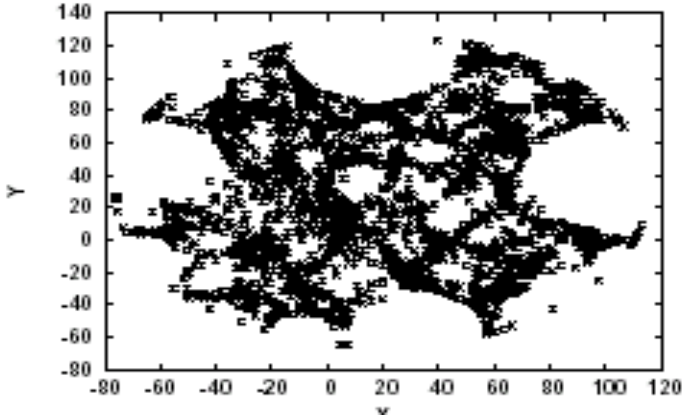


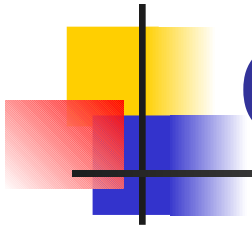
(a)



(b)



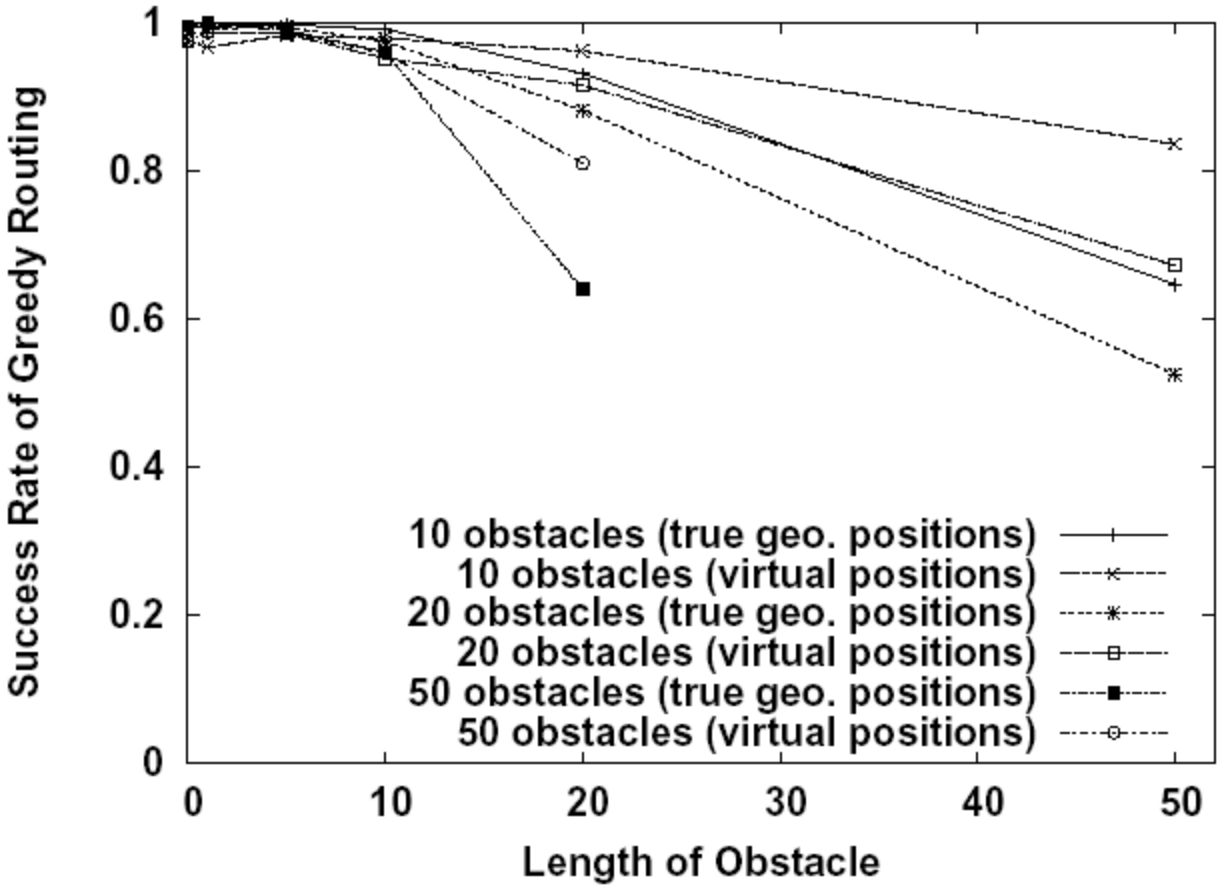


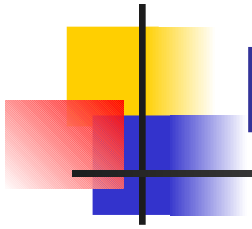


Conclusions

- Geographic routing is useful even without location information
- We can choose coordinates that reflect the true underlying radio connectivity
- Ad-hoc routing can easily scale to tens of thousands of nodes with acceptable overhead
- Future work: ns2, what to do when greedy fails, more DHT studies

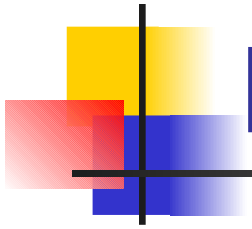
Obstacles


Success Rate of Greedy Routing
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
Length of Obstacle
10 obstacles (true geo. positions)
10 obstacles (virtual positions)
20 obstacles (true geo. positions)
20 obstacles (virtual positions)
50 obstacles (true geo. positions)
50 obstacles (virtual positions)



Recap of the Algorithm - I

- Bootstrap phase
  - Bootstrap node floods
  - Perimeter nodes flood (O(sqrt(N)) overhead, very low constant)
  - Balls and Springs done at each node to fix perimeter nodes



Recap of the Algorithm - I

- Steady state
  - Rubber bands
  - Some designated node floods periodically
    - Need a leader election protocol to deal with failure of this bootstrap node
  - Overhead doesn’t depend on N