Distributed Fine-Grained Node Localization in Ad-Hoc Networks

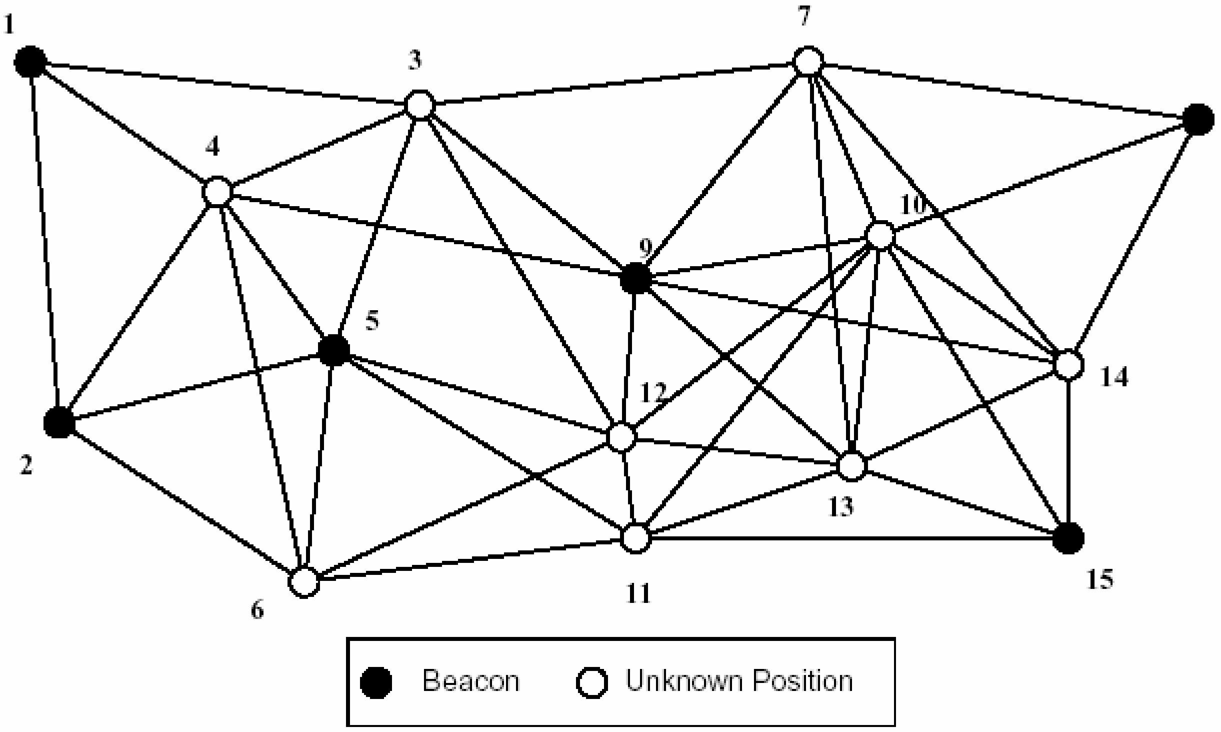
A Scalable Location Service for Geographic Ad Hoc Routing

Presented by An Nguyen

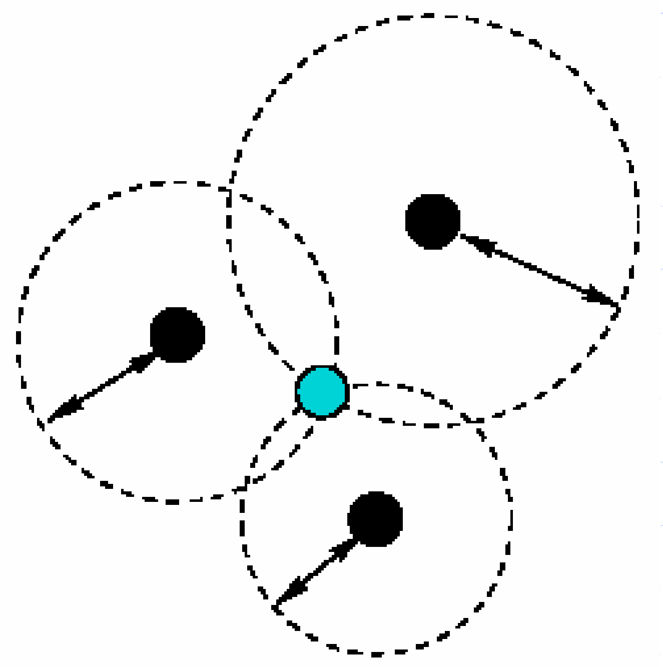
Distributed Fine-Grained Node Localization in Ad-Hoc Networks

By Andreas Savvides
and Mani Srivastava
UCLA

Problem Setting



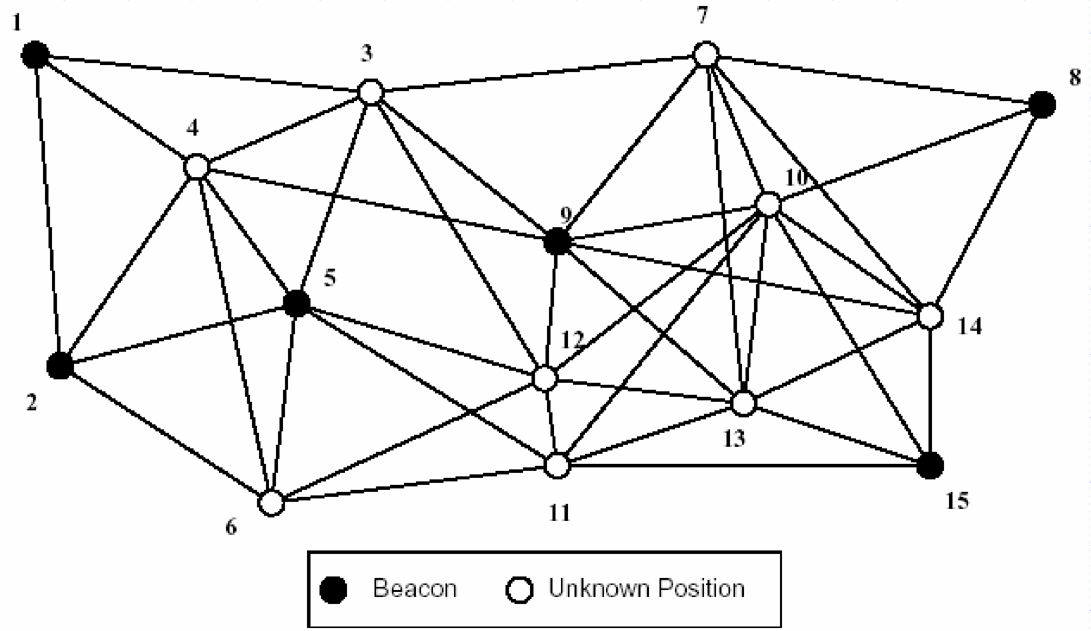
Simple idea: Atomic Multilateration



The position of a node can be determined when 3 beacons are within its range

Slightly more complicated: Iterative Multilateration

- Once a node's position is known, it becomes a beacon

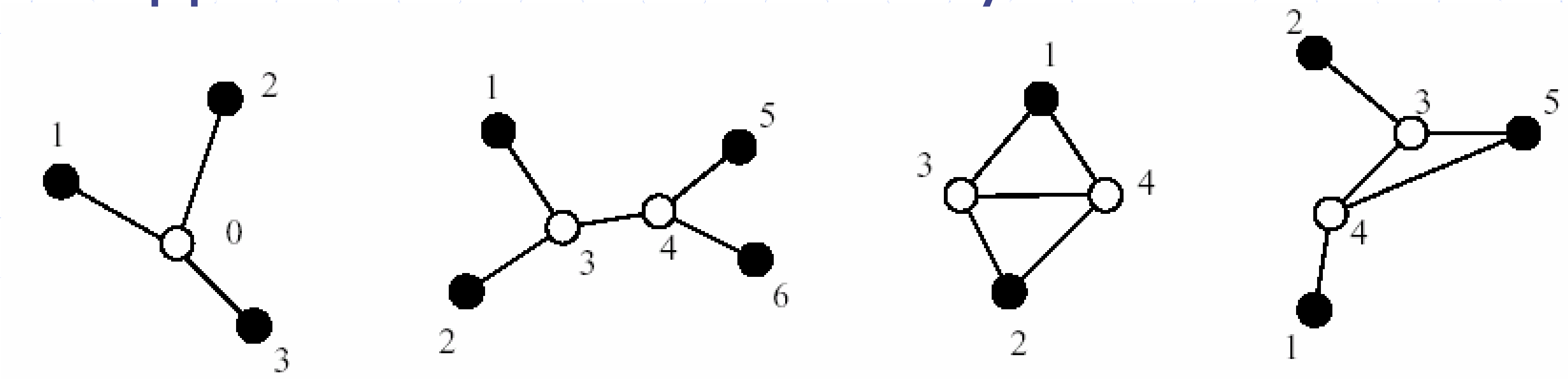


Main Result: Collaborative Multilateration

- Works for more general settings
- 3 phases
  - Formation of collaborative subtrees
  - Computation of initial estimates
  - Position refinement

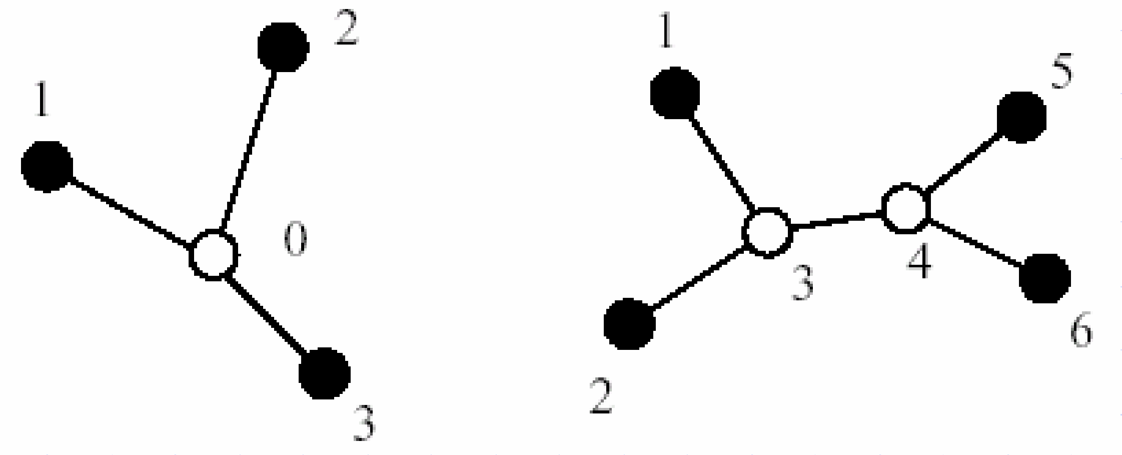
Phase #1: Formation of collaborative subtrees

- Goal:
  - Well-determined or over-determined system of equations
  - Facilitate distributed computation model
- Approach: add nodes one by one



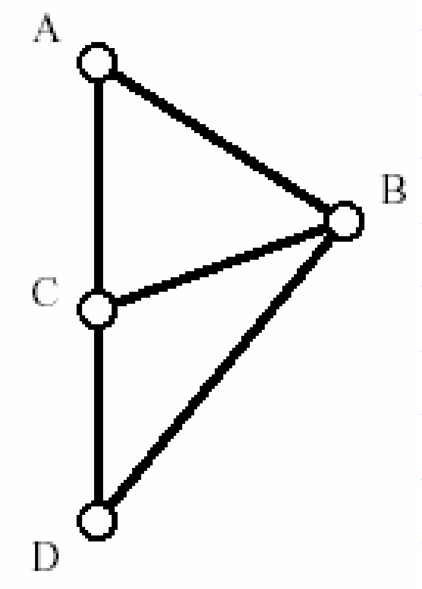
What nodes to add?

- Condition 1: An unknown node that is connected to 3 nodes that are beacons or have tentatively unique position



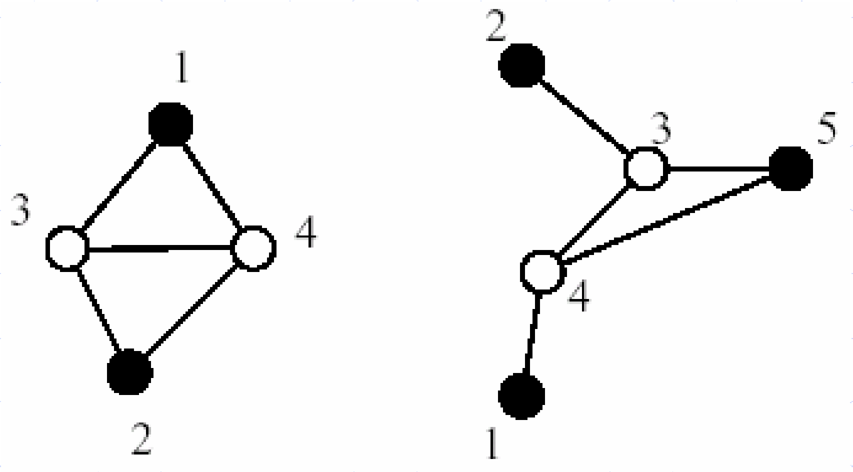
What nodes to add?

- Condition 2: An unknown node uses at least one reference point that is not collinear with the rest of its reference points



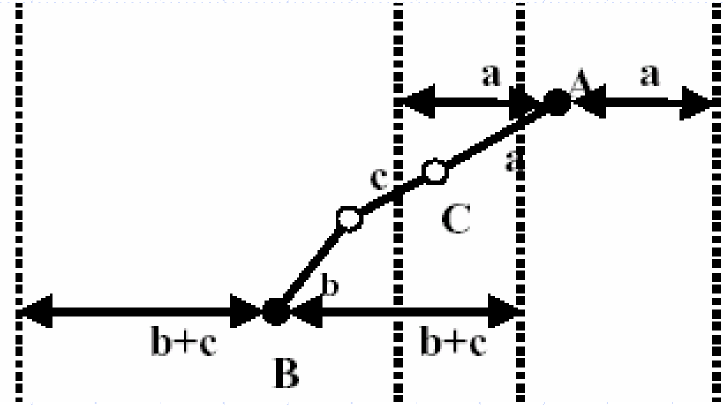
What nodes to add?

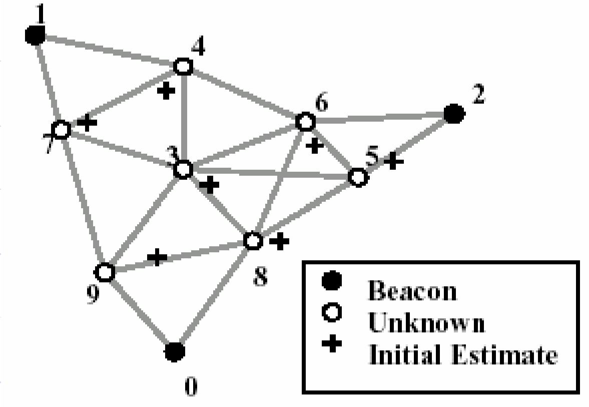
- Condition 3: Each node has at least one link that connects to a different node from the nodes used as references by the other nodes



Phase #2: Computation of initial estimates

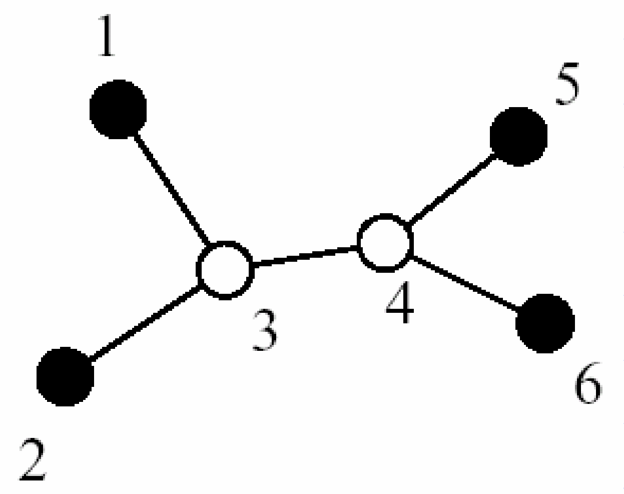
- Find bounding box for each unknown node
- Set initial estimate of the unknown node as the center of its bounding box





Phase #3: Position Refinement (Centralized)

- Minimize the sum of edge error squares
- Use Kalman Filters



$$f_{2,3} = R_{2,3} - \sqrt{(x_2 - ex_3)^2 + (y_2 - ey_3)^2}$$

$$f_{3,5} = R_{3,5} - \sqrt{(ex_3 - x_5)^2 + (ey_3 - y_5)^2}$$

$$f_{4,3} = R_{4,3} - \sqrt{(ex_4 - ex_3)^2 + (ey_4 - ey_3)^2}$$

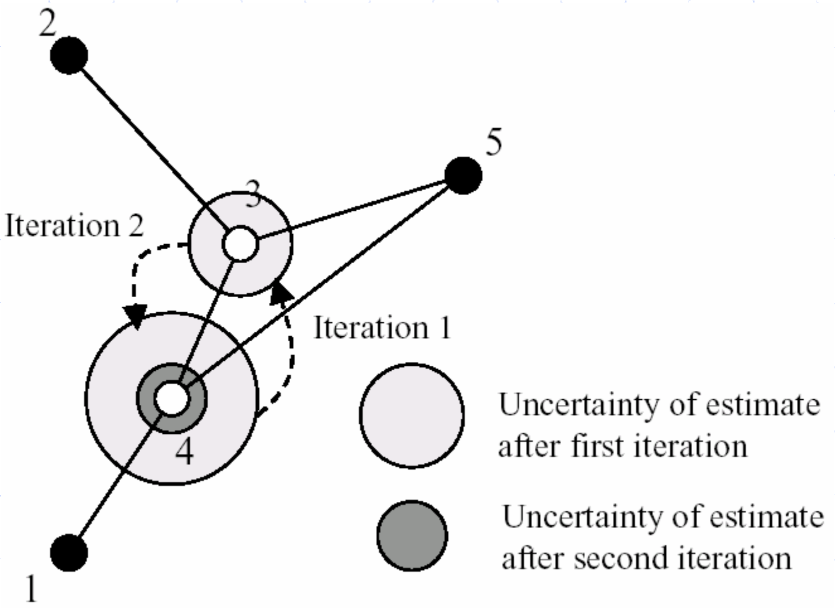
$$f_{4,5} = R_{4,5} - \sqrt{(ex_4 - x_5)^2 + (ey_4 + y_5)^2}$$

$$f_{4,1} = R_{4,1} - \sqrt{(ex_4 - x_1)^2 + (ey_4 - y_1)^2}$$

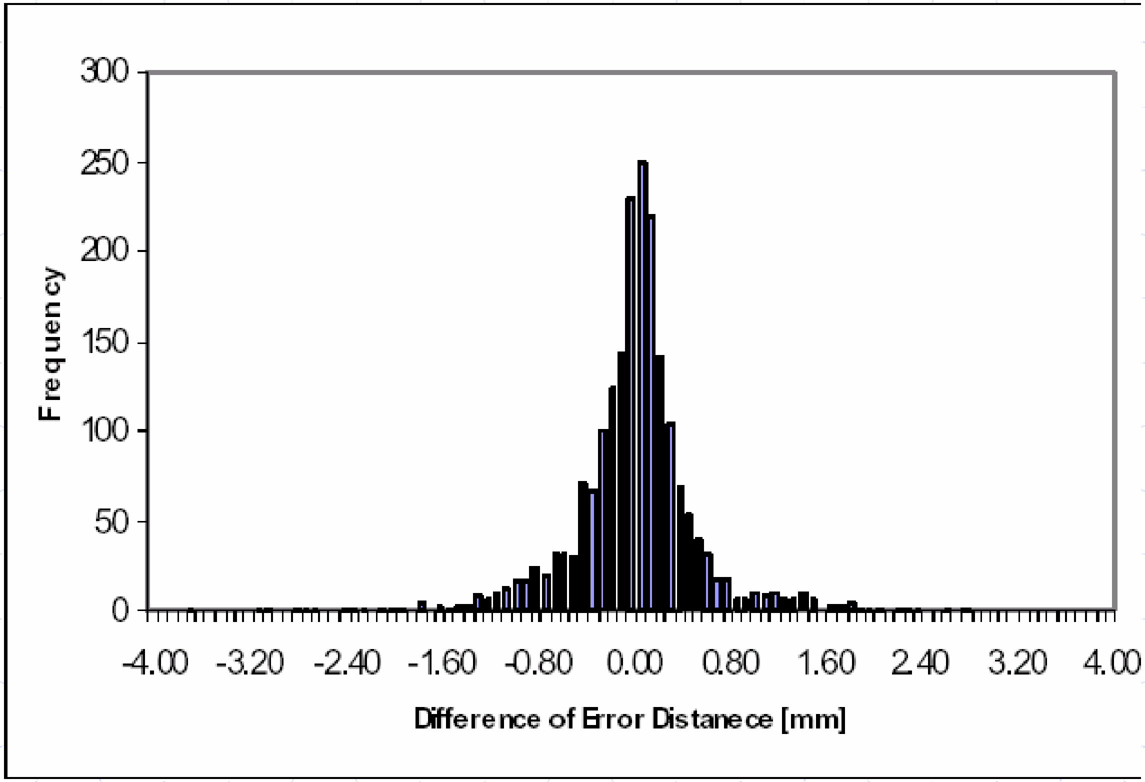
$$F(x_3, y_3, x_4, y_4) = min \sum f_{i,j}^2$$

Phase #3: Position Refinement (Distributed)

- Repeatedly estimate node position using estimated positions of neighbors
- Yield approximately the same result as centralized approach

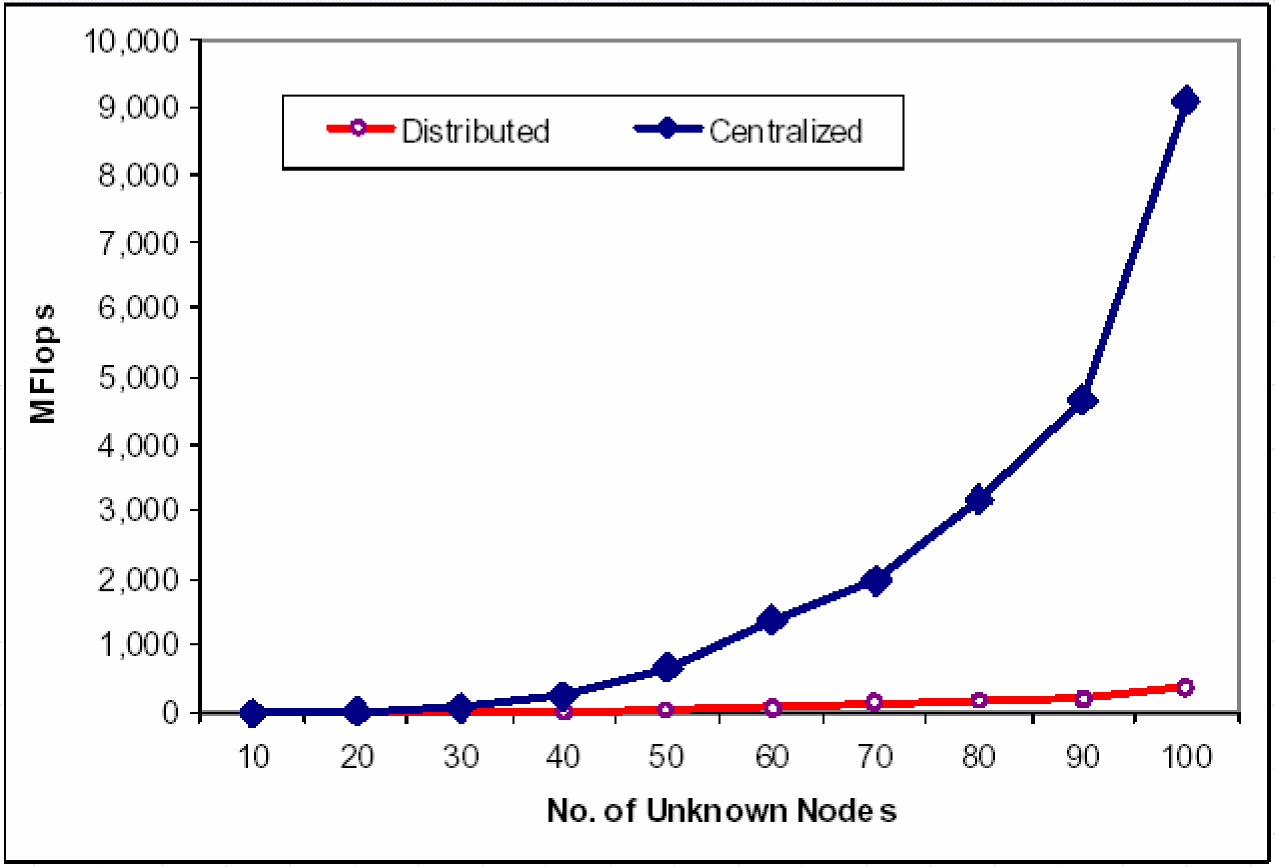

2
5
3
Iteration 2
Iteration 1
4
1
Uncertainty of estimate after first iteration
Uncertainty of estimate after second iteration

Experimental Results (1)



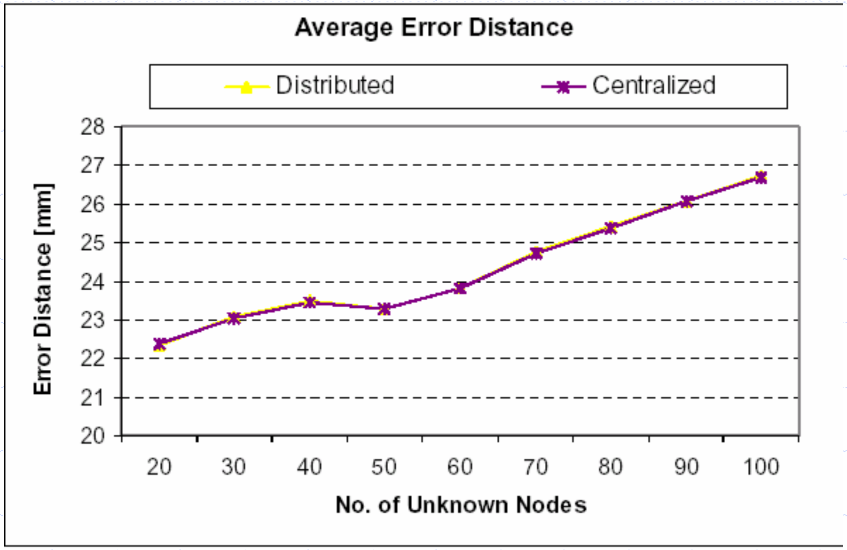
Different between distributed and centralized estimates

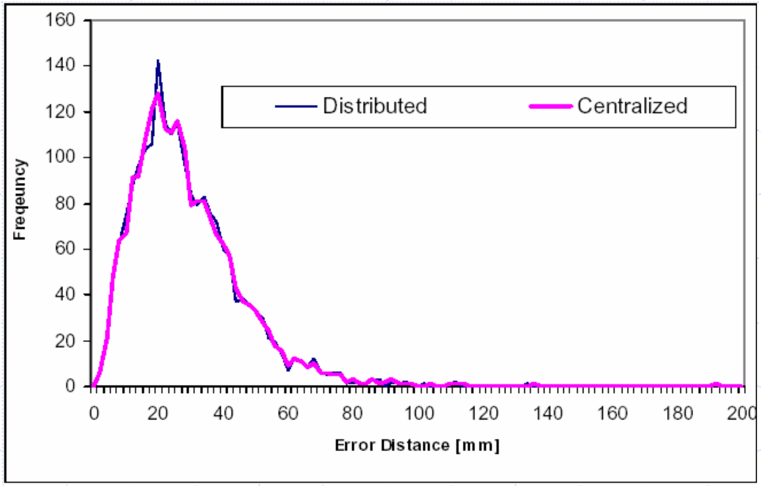
Experimental Results (2)

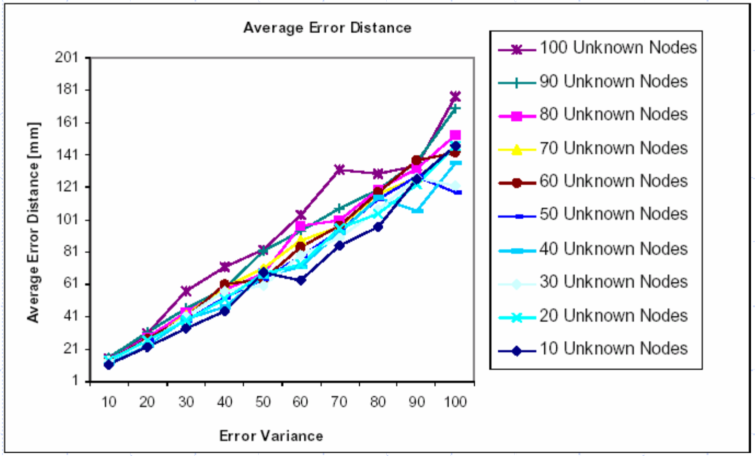


Cost of estimating positions

Experimental Results (3)

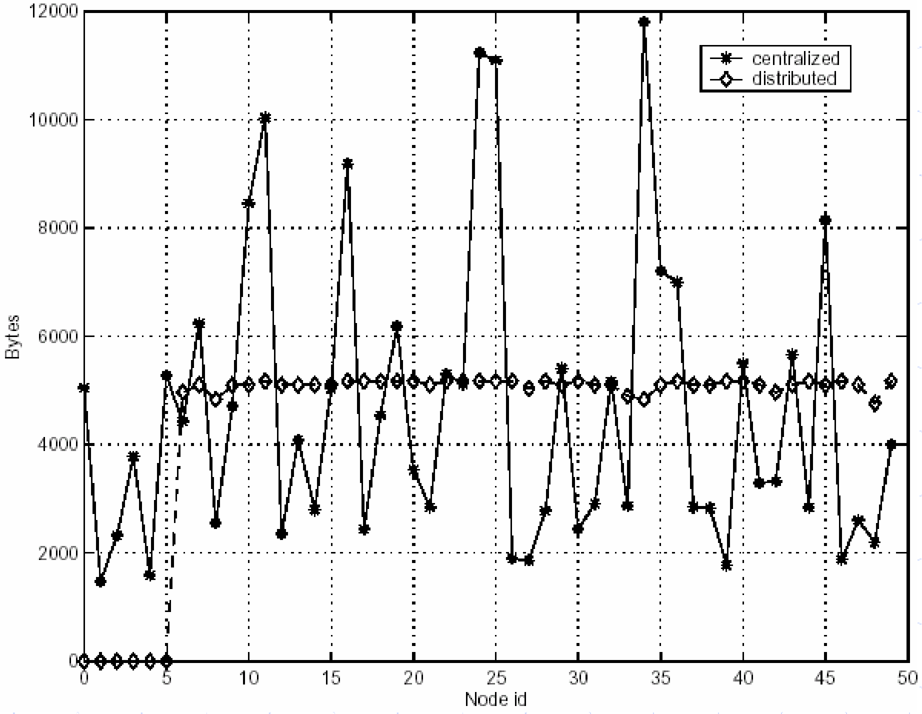




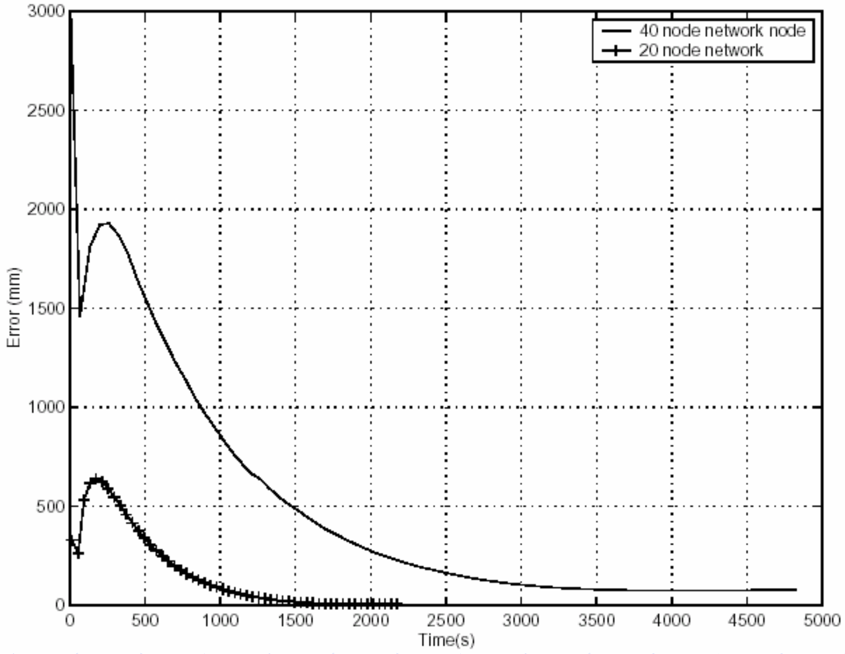


Localization accuracy

Experimental Results (4)



Communication cost



Convergence latency

End of 1st paper

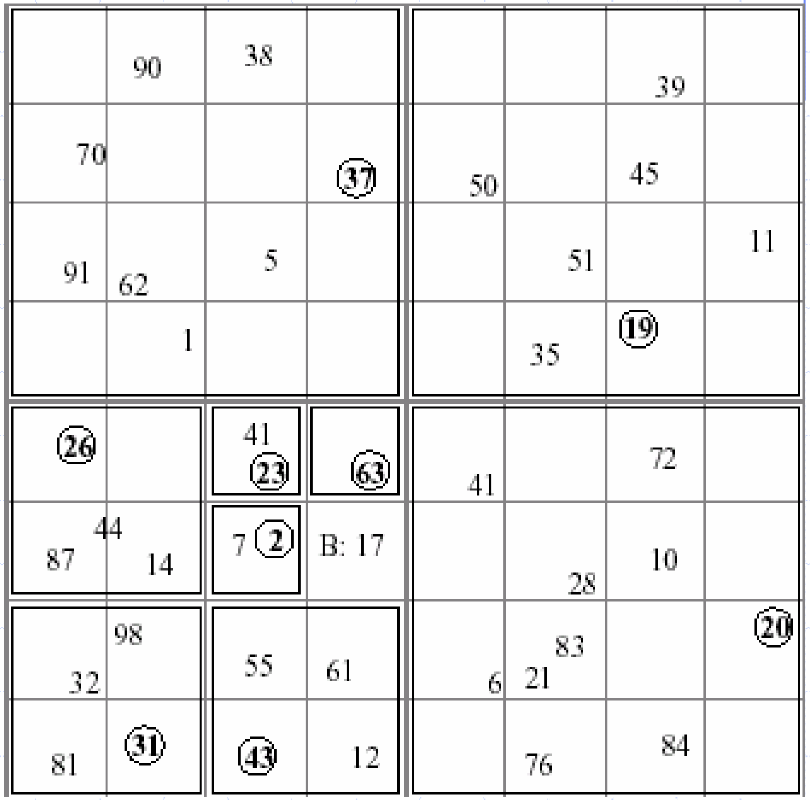
A Scalable Location Service for Geographic Ad Hoc Routing

By Jinyang Li, John Jannotti, Douglas De Couto, David Karger, Robert Morris, MIT

Problem Setting

- Geographic forwarding
- Each node knows its position

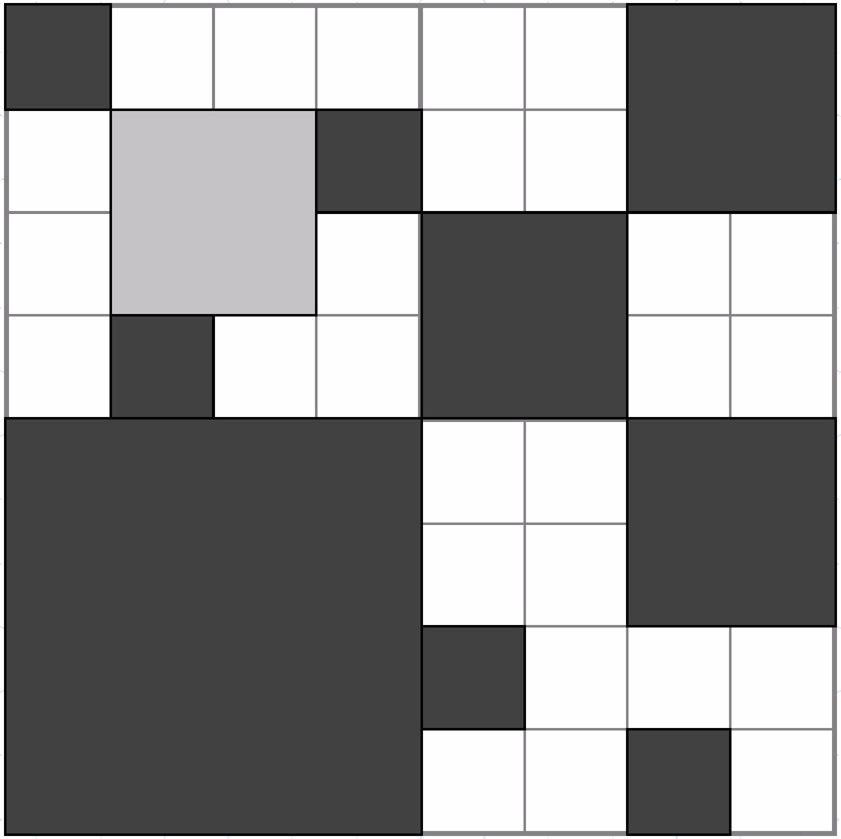
- Location service: given an ID, find the position of a node with that ID



Constraints

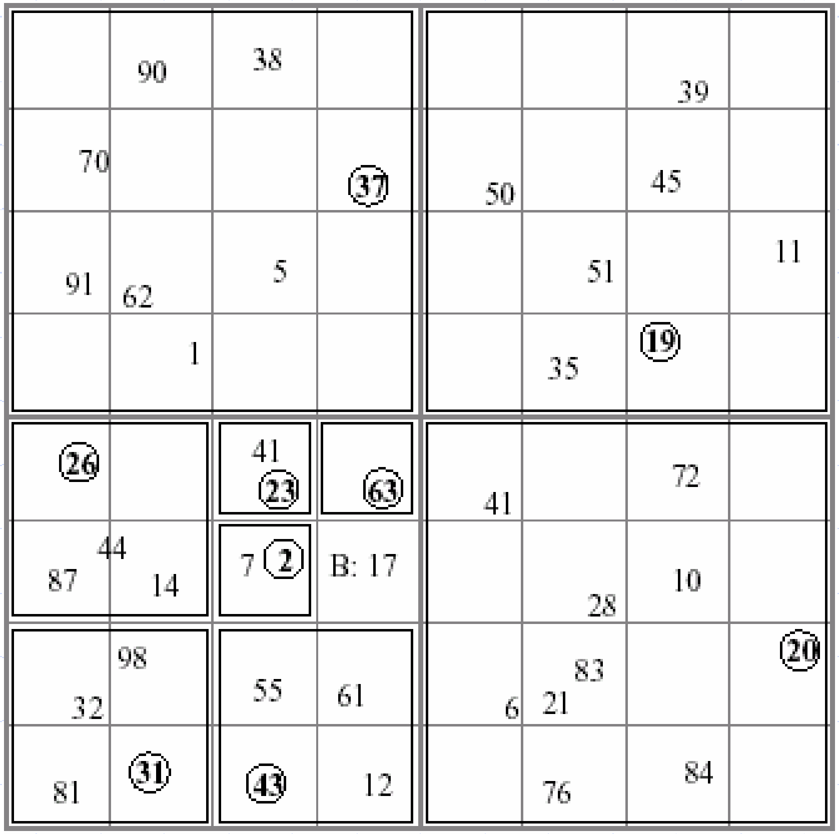
- No node should be a bottle neck,
- Work should be spread evenly
- Failure of a node should not affect much the location service
- Queries for nearby nodes should be local
- Low storage and communication

GLS Idea (1)



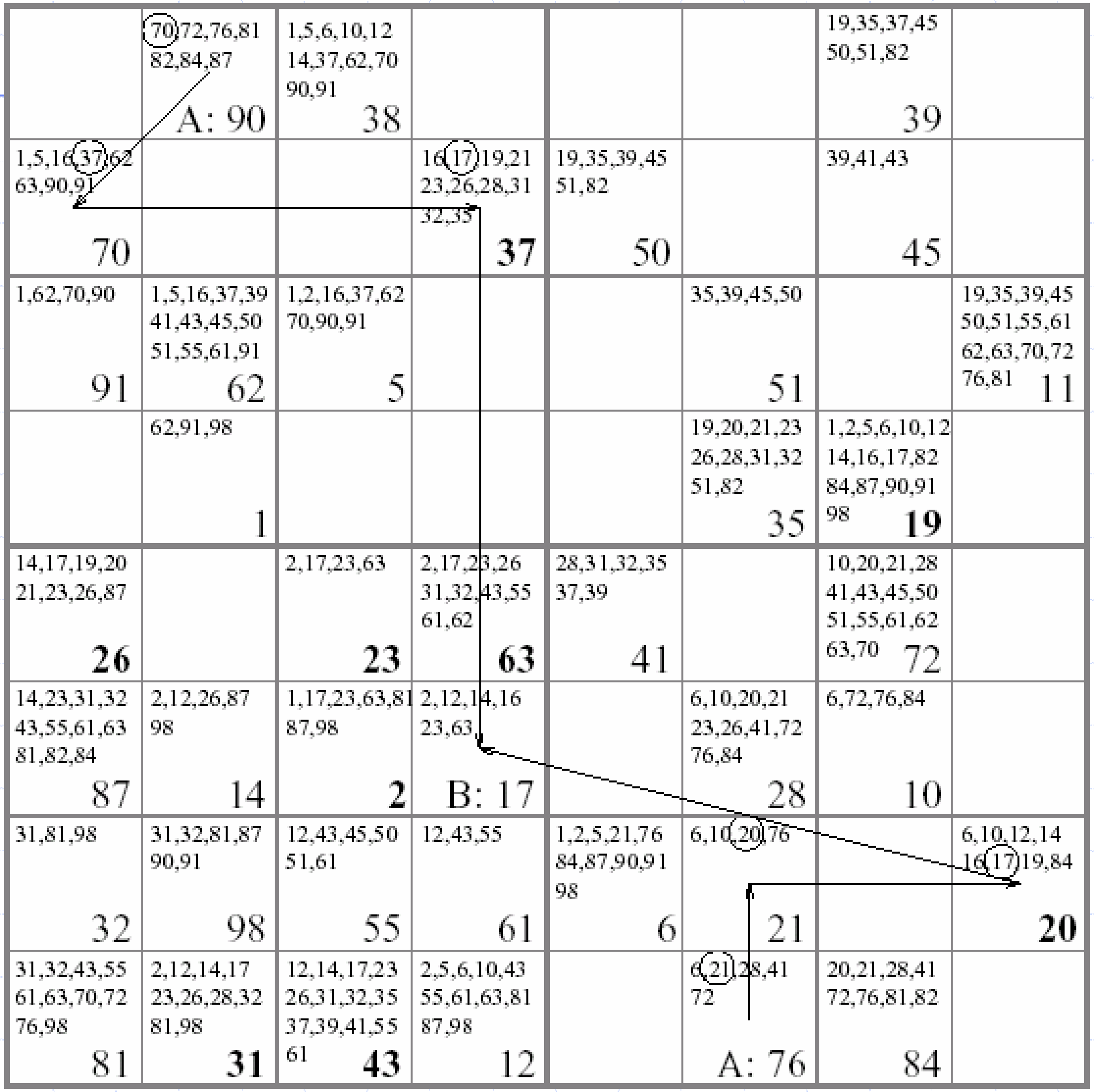
Partition the world

GLS Idea (2)



A node selects location servers “close” to itself
Location servers of a node are well sampled

GLS Query



	70,72,76,81 82,84,87 A: 90	1,5,6,10,12 14,37,62,70 90,91 38				19,35,37,45 50,51,82 39	
1,5,16,37,62 63,90,91 70			16,17,19,21 23,26,28,31 32,35 **37**	19,35,39,45 51,82 50		39,41,43 45	
1,62,70,90 91	1,5,16,37,39 41,43,45,50 51,55,61,91 62	1,2,16,37,62 70,90,91 5			35,39,45,50 51		19,35,39,45 50,51,55,61 62,63,70,72 76,81 11
	62,91,98 1				19,20,21,23 26,28,31,32 51,82 35	1,2,5,6,10,12 14,16,17,82 84,87,90,91 98 **19**	
14,17,19,20 21,23,26,87 **26**		2,17,23,63 **23**	2,17,23,26 31,32,43,55 61,62 **63**	28,31,32,35 37,39 41		10,20,21,28 41,43,45,50 51,55,61,62 63,70 72	
14,23,31,32 43,55,61,63 81,82,84 87	2,12,26,87 98 14	1,17,23,63,81 87,98 **2**	2,12,14,16 23,63 B: 17		6,10,20,21 23,26,41,72 76,84 28	6,72,76,84 10	
31,81,98 32	31,32,81,87 90,91 98	12,43,45,50 51,61 55	12,43,55 61	1,2,5,21,76 84,87,90,91 98 6	6,10,20,76 21		6,10,12,14 16,17,19,84 **20**
31,32,43,55 61,63,70,72 76,98 81	2,12,14,17 23,26,28,32 81,98 **31**	12,14,17,23 26,31,32,35 37,39,41,55 61 **43**	2,5,6,10,43 55,61,63,81 87,98 12		6,21,28,41 72 A: 76	20,21,28,41 72,76,81,82 84	

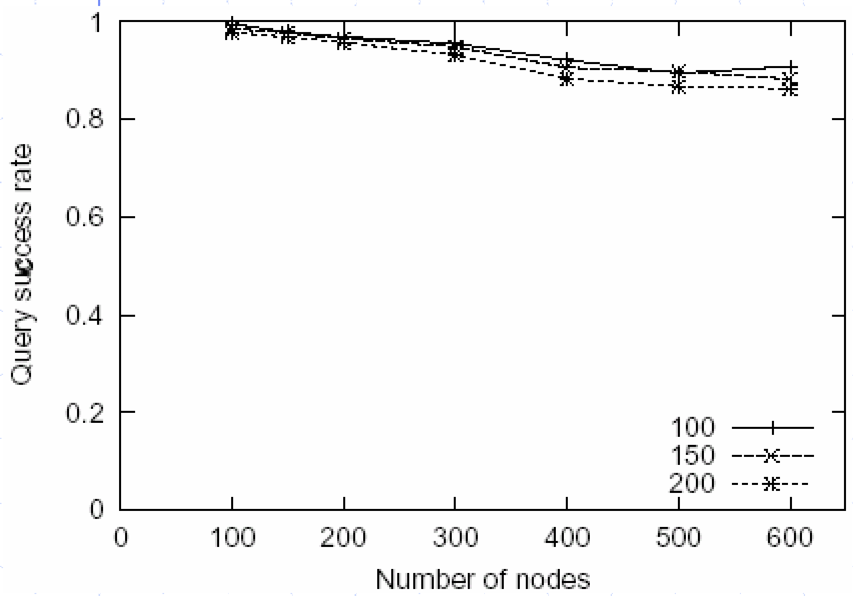
GLS is nice…

- No node is a bottle neck,
- Work is spread evenly
- Failure of a node does not affect much the location service
- Queries for nearby nodes is local
- Low storage and communication

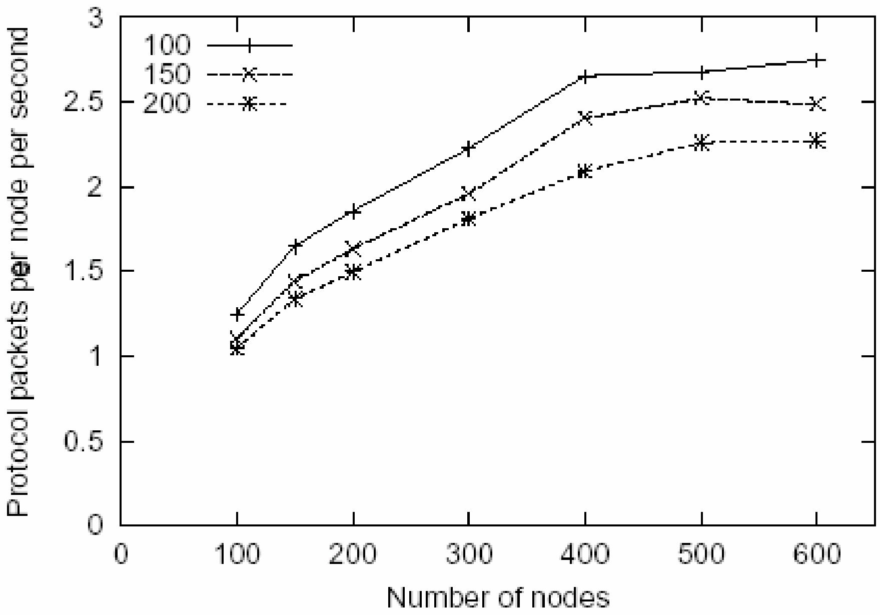
Dealing with motion

- Update its location server from time to time
  - Higher level location server are updated less frequently

Experimental Results (1)

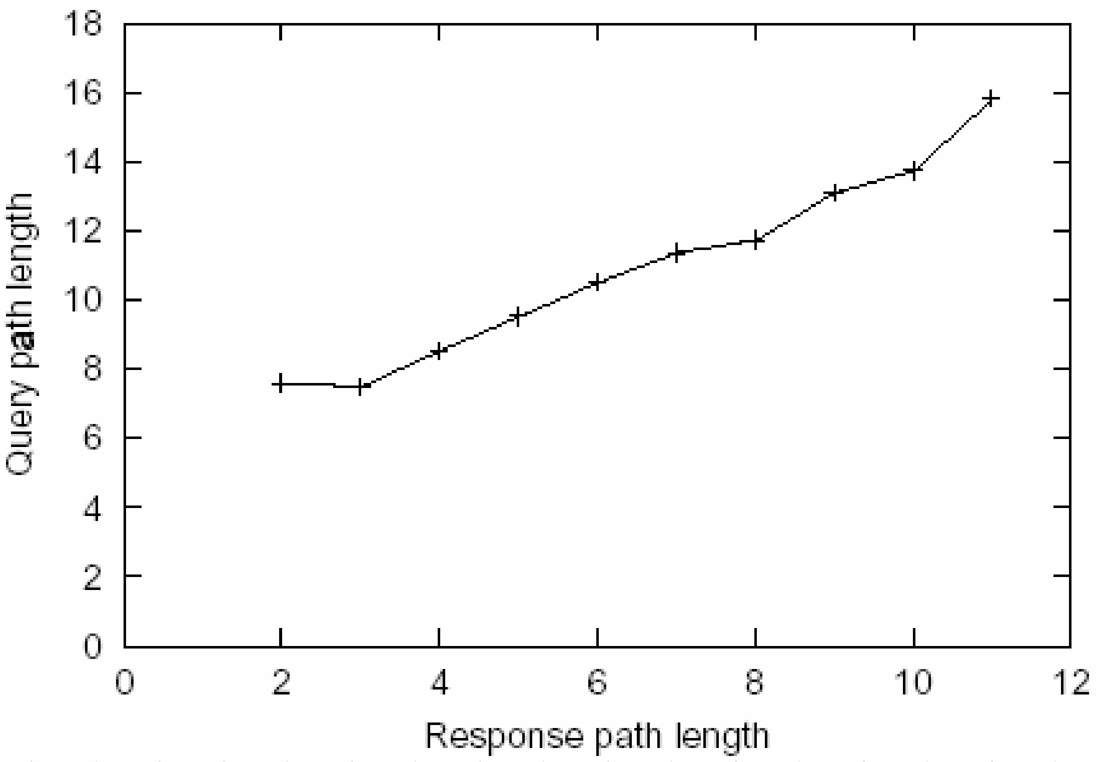


Query success rate



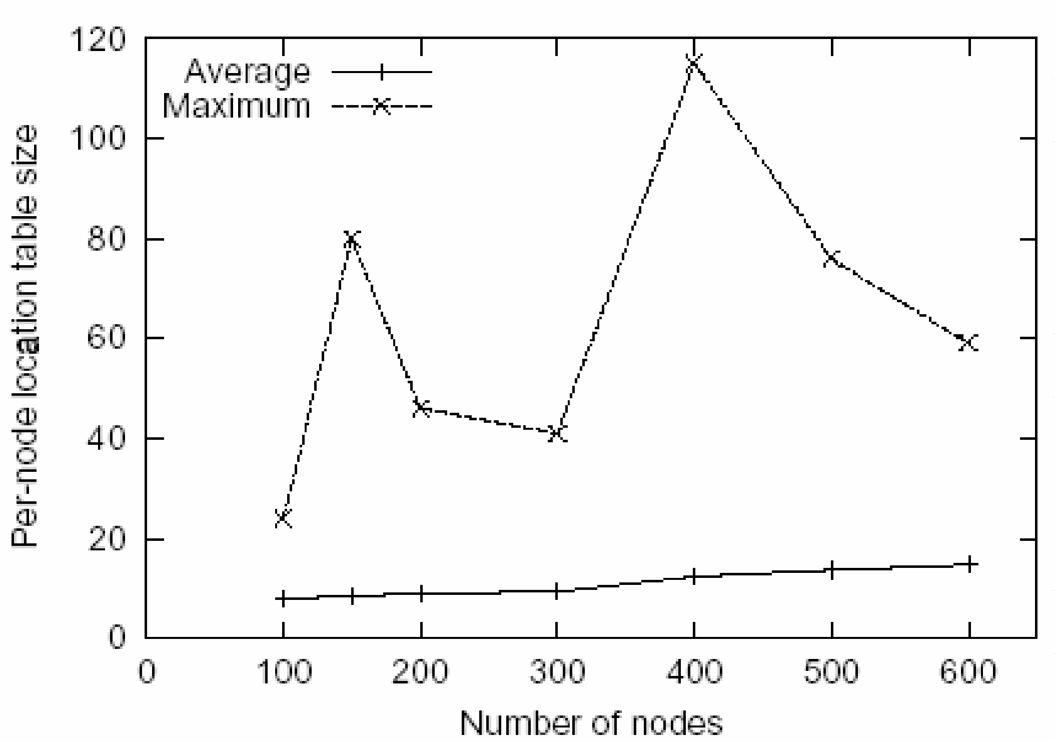
Number of packets passing through a node

Experimental Results (2)



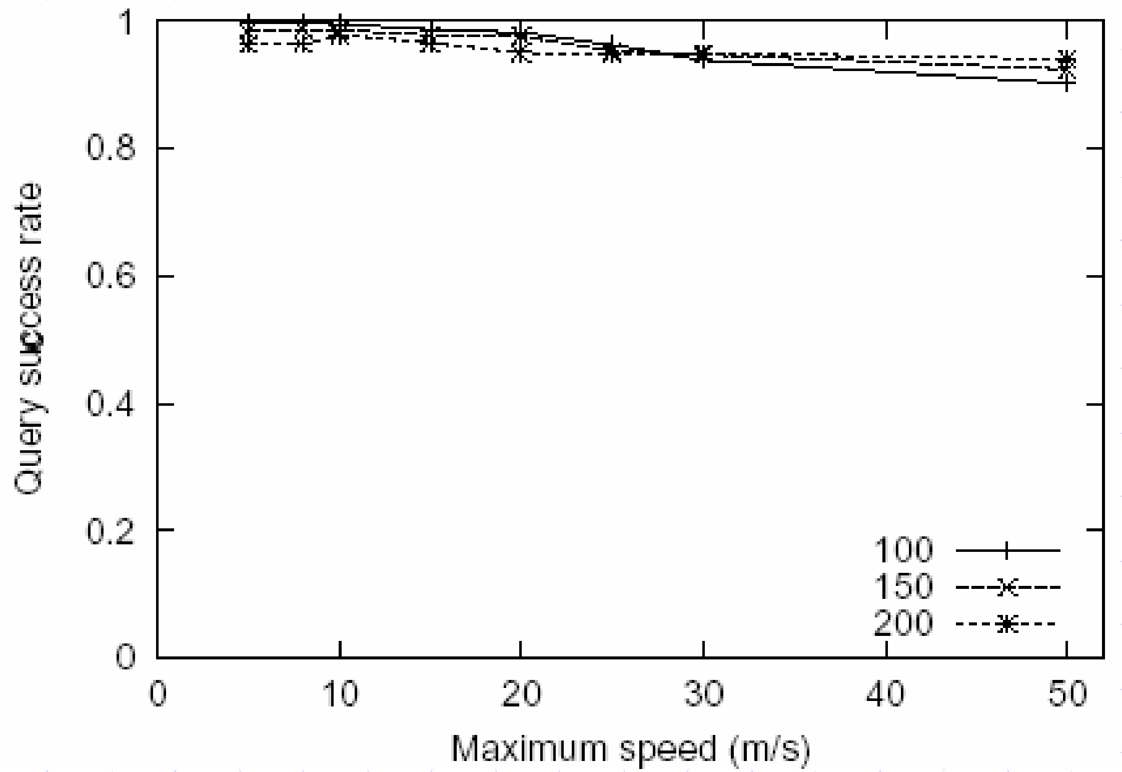
Query path vs communication path

Experimental Results (3)



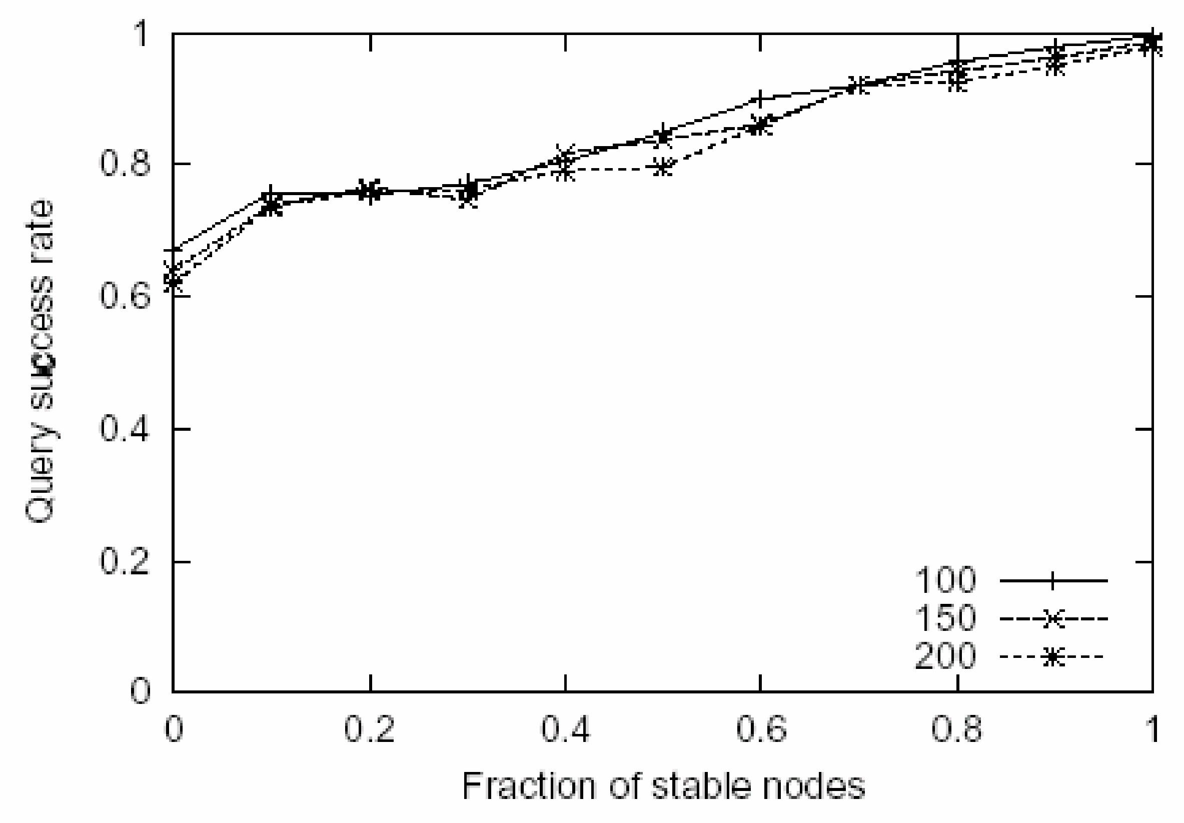
Storage per node

Experimental Results (4)



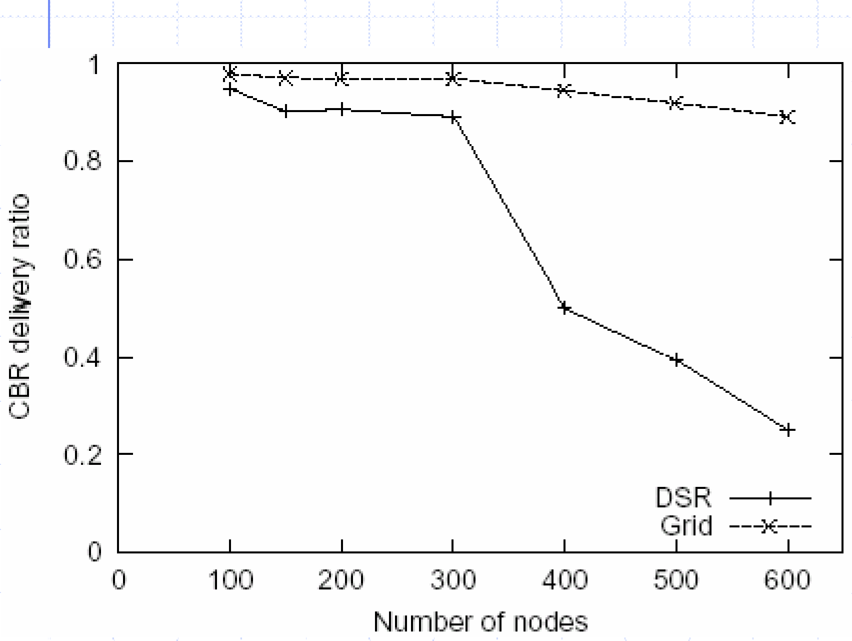
Query success rate vs node speed

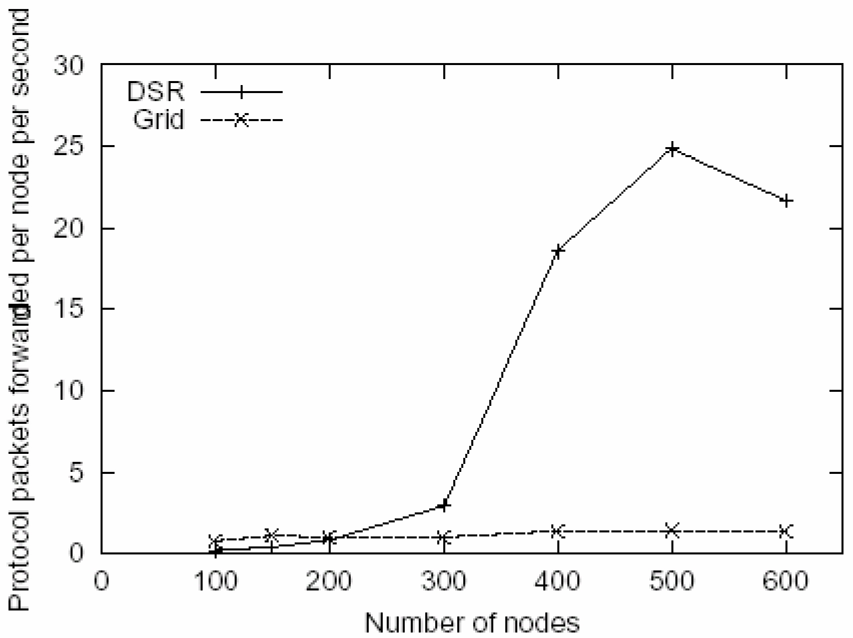
Experimental Results (5)



Query success rate vs node failure rate

Experimental Results (6)





Delivery rate

Packets per node

GLS + Data traffic

Summary

- Localization
  - Distributed
  - Accurate
- Location Service for Geographic Forwarding
  - Local
  - Balanced
  - No bottle neck nodes
  - Handle node failures gracefully
  - Low storage/bandwidth requirement