Localization in Sensor Networks

4/3/06

Some slides are made by Savvides

Find where the sensor is…

- Location information is important.
  1. Devices need to know where they are.
     - Sensor tasking: turn on the sensor near the window…
  2. We want to know where the data is about.
     - A sensor reading is too hot – where?
  3. It helps infrastructure establishment, such as geographical routing and sensor coverage.
     - Intruder detection;
     - Localized geographical routing.

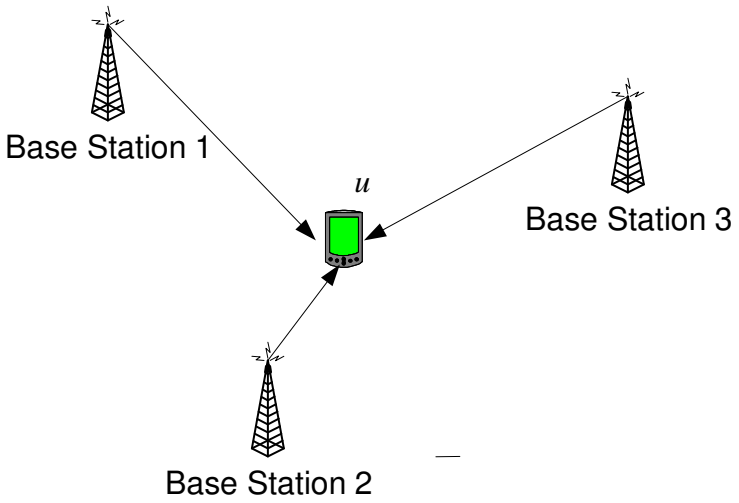
Papers

- **Multi-lateration:**

- **[Savvides01]** A. Savvides, C.-C. Han, and M. B. Strivastava. **Dynamic fine-grained localization in ad-hoc networks of sensors**. Proc. MobiCom 2001.
- **[Savvides03]** A. Savvides, H. Park, and M. B. Strivastava. **The *n*-hop multilateration primitive for node localization problems**, Mobile Networks and Applications, Volume 8, Issue 4, 443-451, 2003.

Multilateration: use plane geometry

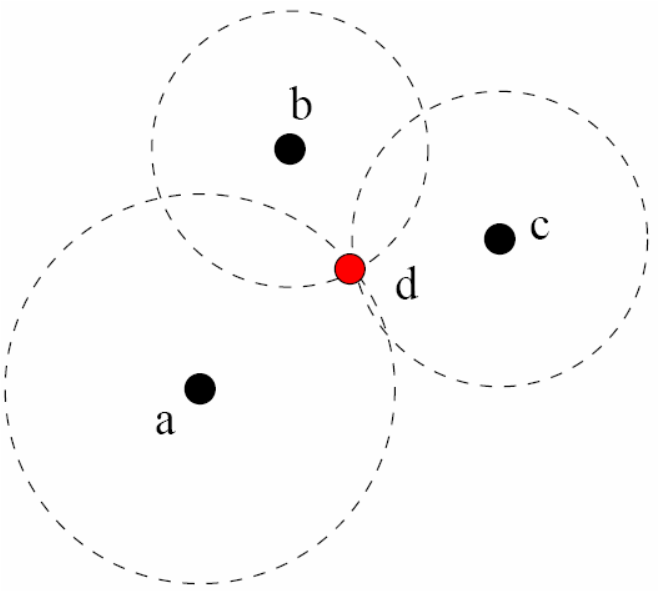
Base Case: Atomic Multilateration

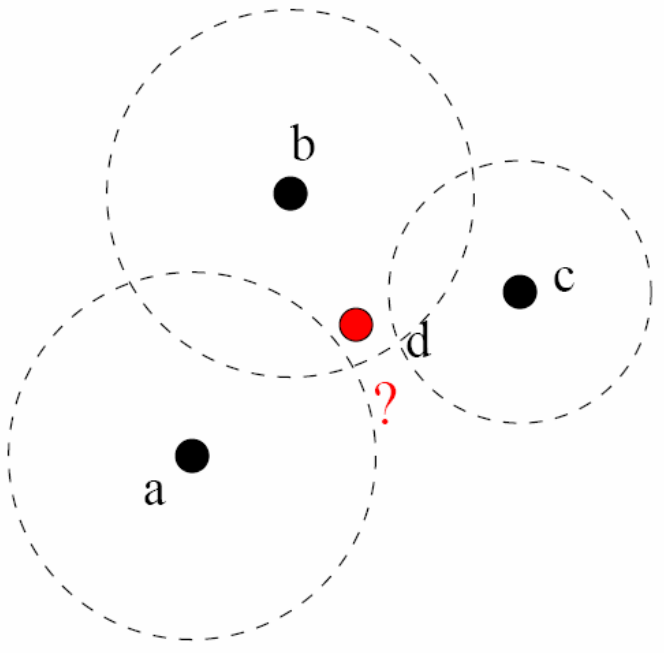


- Base stations advertise their coordinates & transmit a reference signal
- PDA uses the reference signal to estimate distances to each of the base stations

Base Case: Atomic Multilateration

- Distance measurements are noisy!
- Solve an optimization problem: minimize the mean square error.





Problem Formulation

- k beacons at positions (x_i, y_i)
- Assume node 0 has position (x_0, y_0)
- Distance measurement between node 0 and beacon i is r_i
- Error:

$$f_i = r_i - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$$

- The objective function is

$$F(x_0, y_0) = \min \sum f_i^2$$

- This is a non-linear optimization problem

Linearization and Min Mean Square Estimate

- Ideally, we would like the error to be 0

$$f_i = r_i - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} = 0$$

- Re-arrange:

$$(x_0^{\,2} + y_0^{\,2}) + x_0(-2x_i) + y_0(-2y_i) - r_i^{\,2} = -x_i^2 - y_i^2$$

- Subtract the last equation from the previous ones to get rid of quadratic terms.

$$2x_0(x_k - x_i) + 2y_0(y_k - y_i) = r_i^2 - r_k^{\,2} - x_i^2 - y_i^2 + x_k^2 + y_k^2$$

- Note that this is linear.

Linearization and Min Mean Square Estimate

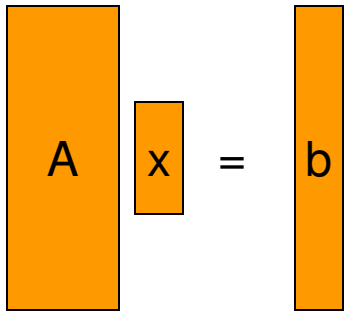
- In general, we have an over-constrained linear system

$$Ax = b$$

$$b = \begin{bmatrix} r_1^2 - r_k^2 - x_1^2 - y_1^2 + x_k^2 + y_k^2 \\ r_2^2 - r_k^2 - x_2^2 - y_2^2 + x_k^2 + y_k^2 \\ \vdots \\ r_{k-1}^2 - r_k^2 - x_{k-1}^2 - y_{k-1}^2 + x_k^2 + y_k^2 \end{bmatrix}$$

$$A = \begin{bmatrix} 2(x_k - x_1) & 2(y_k - y_1) \\ 2(x_k - x_2) & 2(y_k - y_2) \\ \vdots & \vdots \\ 2(x_k - x_{k-1}) & 2(y_k - y_{k-1}) \end{bmatrix}$$

$$x = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$



Solve using the Least Square Equation

The linearized equations in matrix form become

$$Ax = b$$

Now we can use the least squares equation to compute an estimation.

$$x = (A^T A)^{-1} A^T b$$

How to solve it in an embedded system?

- Check conditions
  - Beacon nodes must not lie on the same line
- For ToA, TDoA, how to solve for the speed of sound?

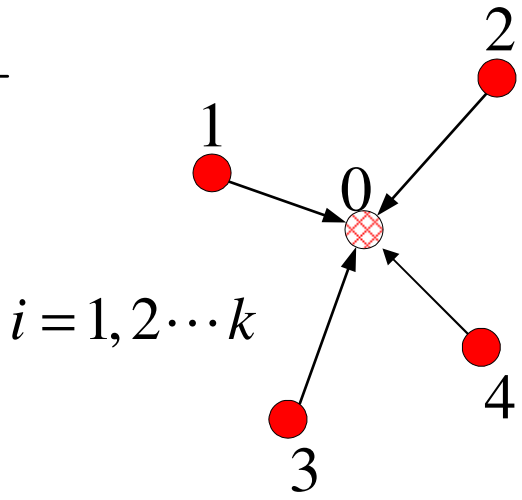
Acoustic case: Also solve for the speed of sound

With at least 4 beacons,

$$f_i = st_{i0} - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$$

Speed of sound

Time measurement



This can be linearized to the form

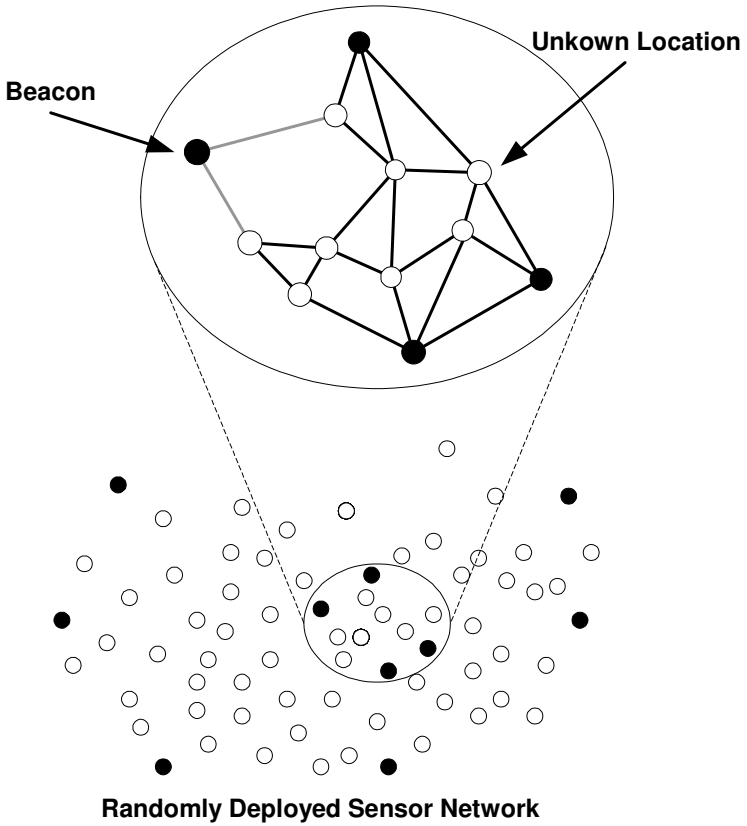
$$Ax = b$$

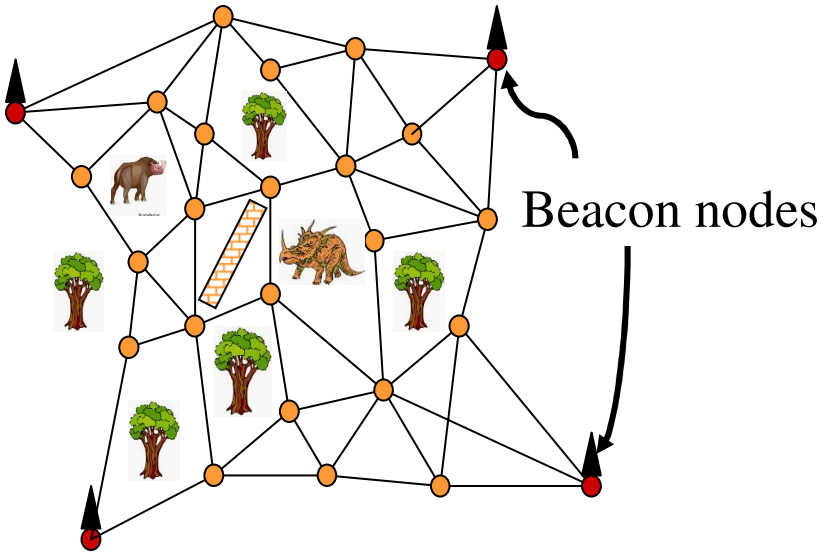
where

$$b = \begin{bmatrix} -x_1^2 - y_1^2 + x_k^2 + y_k^2 \\ -x_2^2 - y_2^2 + x_k^2 + y_k^2 \\ \vdots \\ -x_{k-1}^2 - y_{k-1}^2 + x_k^2 + y_k^2 \end{bmatrix} \quad A = \begin{bmatrix} 2(x_k - x_1) & 2(y_k - y_1) & t_{k0}^2 - t_{10}^2 \\ 2(x_k - x_2) & 2(y_k - y_2) & t_{k0}^2 - t_{20}^2 \\ \vdots & \vdots & \\ 2(x_k - x_{k-1}) & 2(y_k - y_{k-1}) & t_{k0}^2 - t_{(k-1)0}^2 \end{bmatrix} \quad x = \begin{bmatrix} x_0 \\ y_0 \\ s^2 \end{bmatrix}$$

MMSE Solution: $x = (A^T A)^{-1} A^T b$

The Node Localization Problem

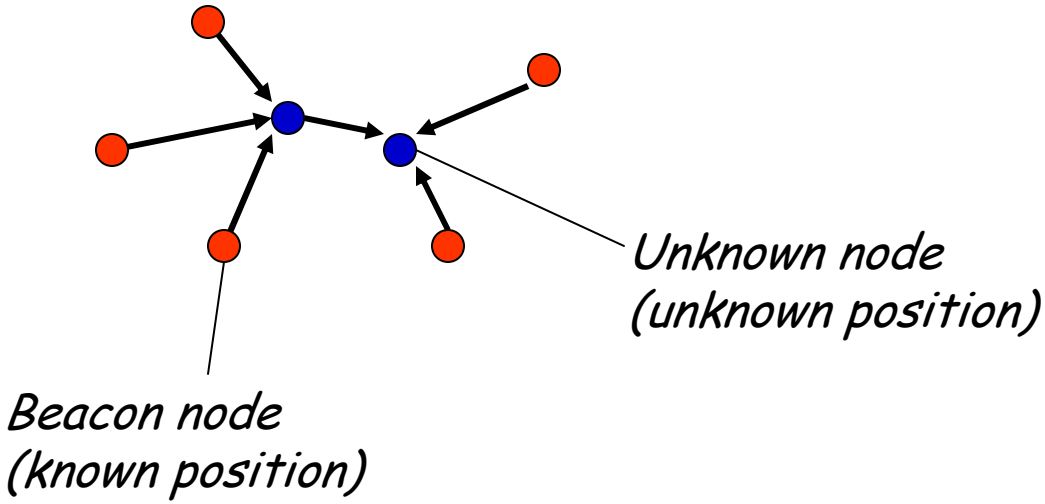




- Localize nodes in an ad-hoc multihop network
- Based on a set of inter-node distance measurements

Solving over multiple hops

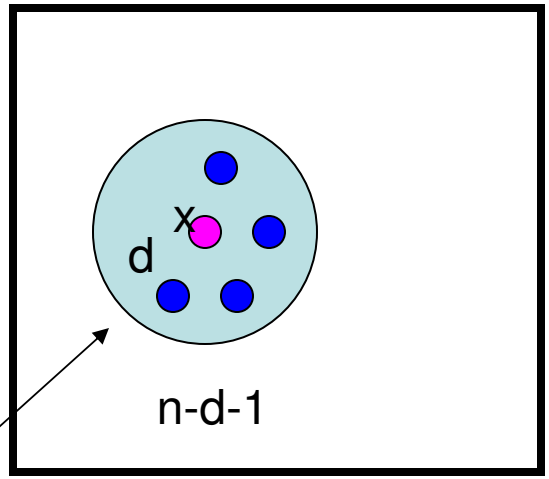
- Iterative multilateration
  - a node with at least 3 neighboring beacons estimates its position and becomes a beacon.
  - Iterate until all nodes with 3 beacons are localized.



Connectivity matters! Each node needs at least 3 neighbors.

Iterative multilateration: how many beacons?

- n nodes deployed randomly in a square of side L,
- P(d)=Pr{a node x has degree d}=?



Transmission range has radius R

Probability that one node falls inside the transmission range of x?

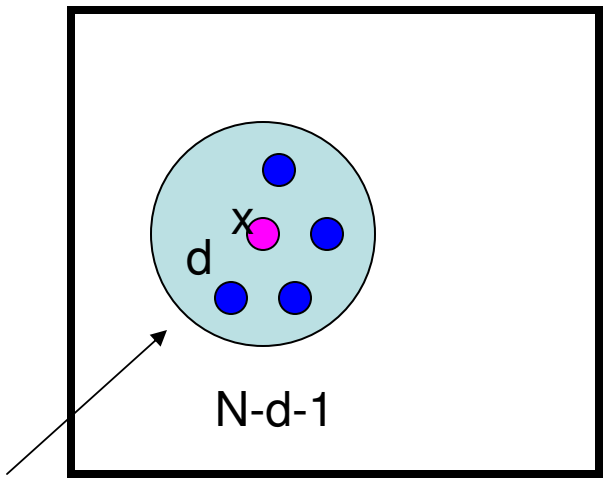
$$p = \frac{\pi R^2}{L^2}$$

Binomial distribution

$$P(d) = p^d \cdot (1-p)^{n-d-1} \cdot \binom{n-1}{d}$$

Iterative multilateration: how many beacons?

- When n tends to infinity, the binomial distribution converges to a Poisson distribution.



Transmission range has radius R

Probability that one node falls inside the transmission range of x?

$$p = \frac{\pi R^2}{L^2} \qquad \lambda = n \cdot p$$

↓ Binomial distribution
Poisson distribution

$$P(d) = \frac{\lambda^d}{d!} \cdot e^{-\lambda}$$

Iterative multilateration: how many beacons?

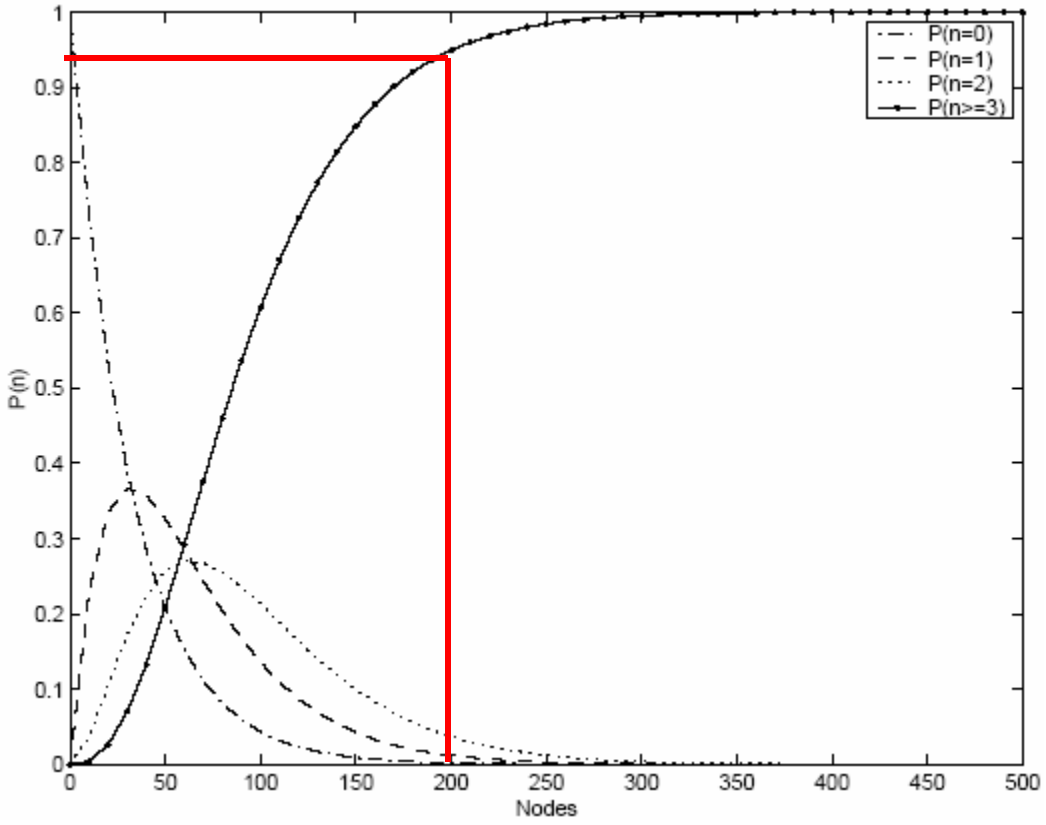
$$P(d) = \frac{\lambda^d}{d!} \cdot e^{-\lambda}$$

$$P(\geq d) = 1 - \sum_{i=1}^{n-1} P(i)$$

100 by 100 field
Sensor range:10

Probability of a node with 0, 1, 2, ≥ 3 neighbors.

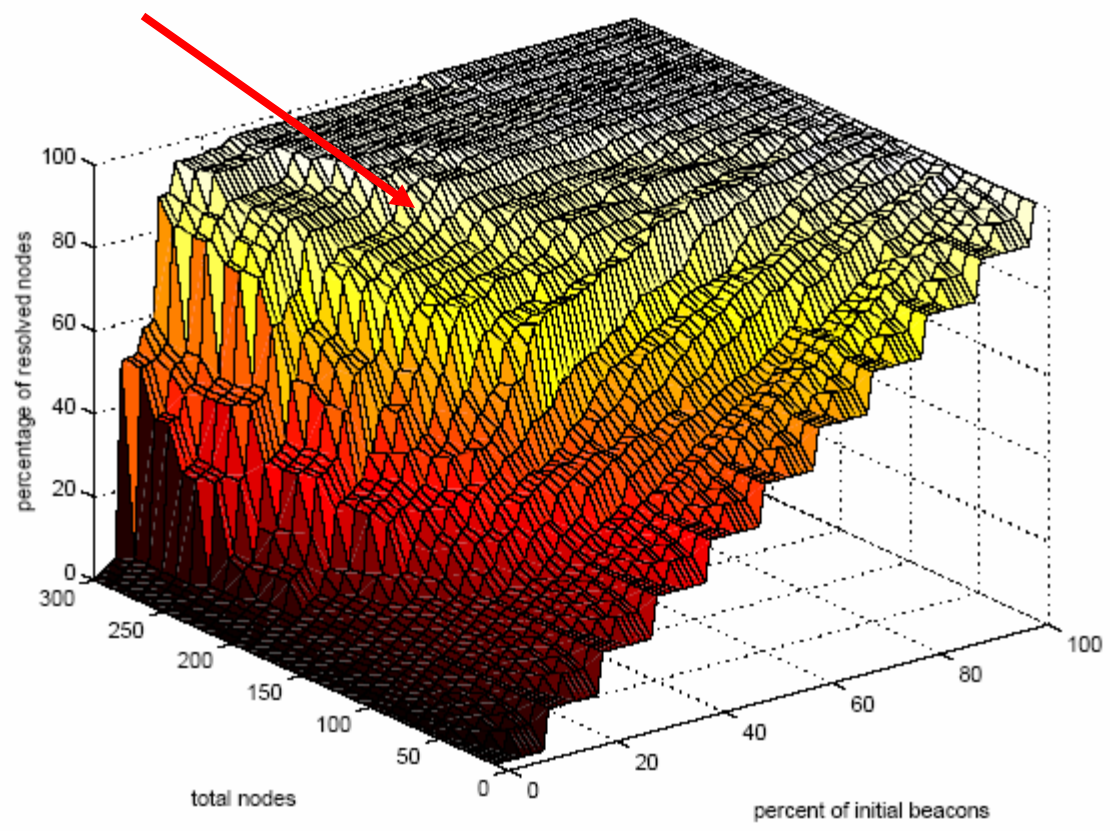
With 200 nodes, P(≥ 3) is about 95%.



Iterative multilateration: how many beacons?

With 200 nodes, P(≥ 3) is about 95%.

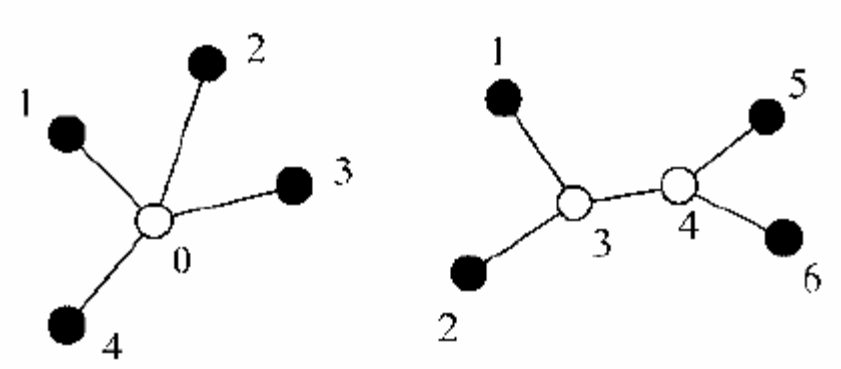
With 200 nodes, we need about 50~60 beacons to localize about 90% of the nodes. That's ¼ of the total number of nodes.



Problems of iterative Multilateration

Problems

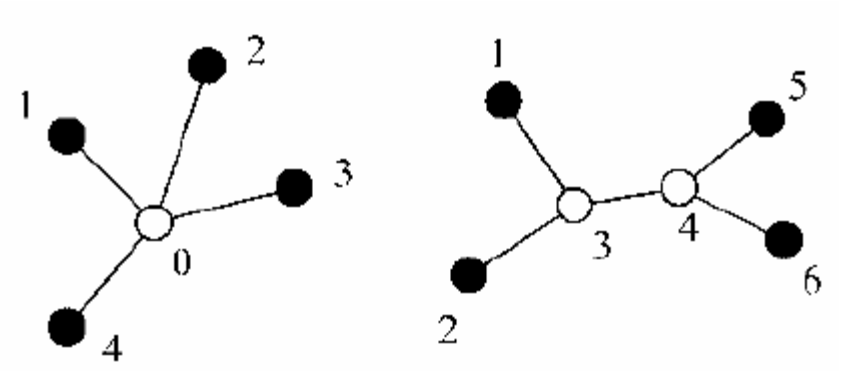
1. Requires a large fraction of beacons.
2. Error accumulates.
3. It gets stuck --- not all nodes with 3 or more neighbors can be resolved.



Problems of iterative Multilateration

Problems

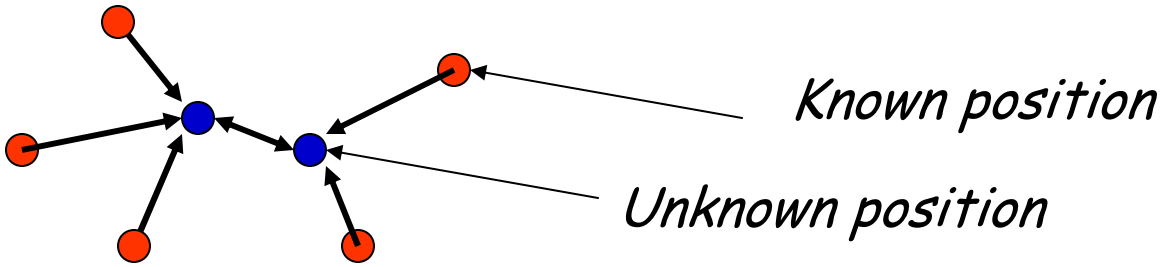
1. Requires a large fraction of beacons.
2. Error accumulates. ← Mass-spring optimization.
3. It gets stuck --- not all nodes with 3 or more neighbors can be located. ← Collaborative multilateration



Collaborative Multilateration: use joint optimization

Collaborative Mutlilateration

– All available measurements are used as constraints

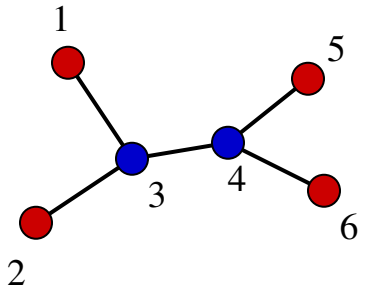


– Solve for the positions of multiple unknowns simultaneously

– Joint optimization can get better results compared with separate optimizations.

Problem Formulation

$$f_{2,3} = r_{2,3} - \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2}$$
$$f_{3,5} = r_{3,5} - \sqrt{(x_3 - x_5)^2 + (y_3 - y_5)^2}$$
$$f_{4,3} = r_{4,3} - \sqrt{(x_4 - x_3)^2 + (y_4 - y_3)^2}$$
$$f_{4,5} = r_{4,5} - \sqrt{(x_4 - x_5)^2 + (y_4 - y_5)^2}$$
$$f_{4,1} = r_{4,1} - \sqrt{(x_4 - x_1)^2 + (y_4 - y_1)^2}$$



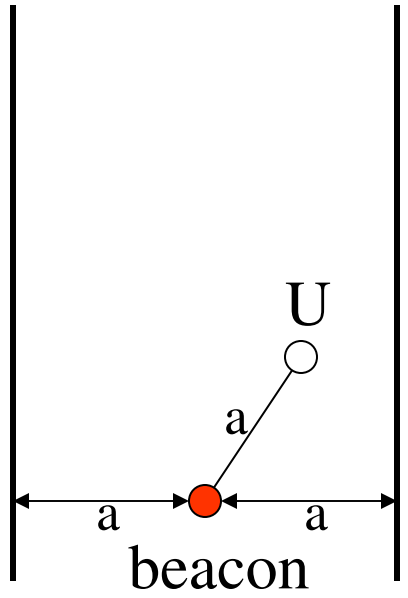
The objective function is

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

Start from some initial estimates, then use a Kalman Filter.

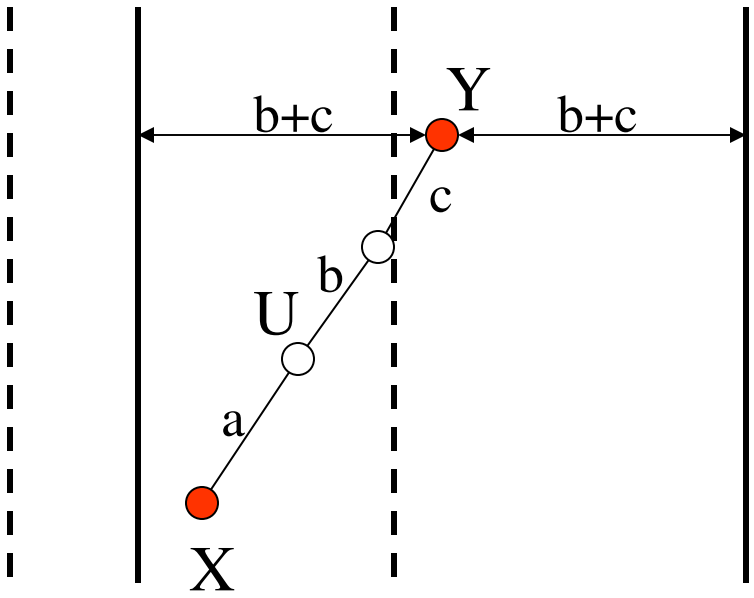
Initial Estimates

- **Use the distance to a beacon as bounds on the x and y coordinates**



Initial Estimates (Phase 2)

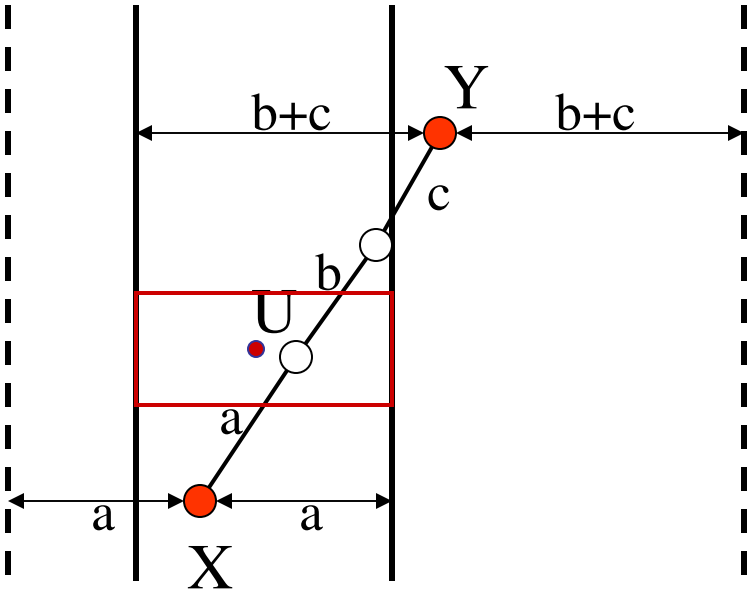
- **Use the distance to a beacon as bounds on the x and y coordinates**
- **Do the same for beacons that are multiple hops away**
- **Select the most constraining bounds**



U is between [Y-(b+c)] and [X+a]

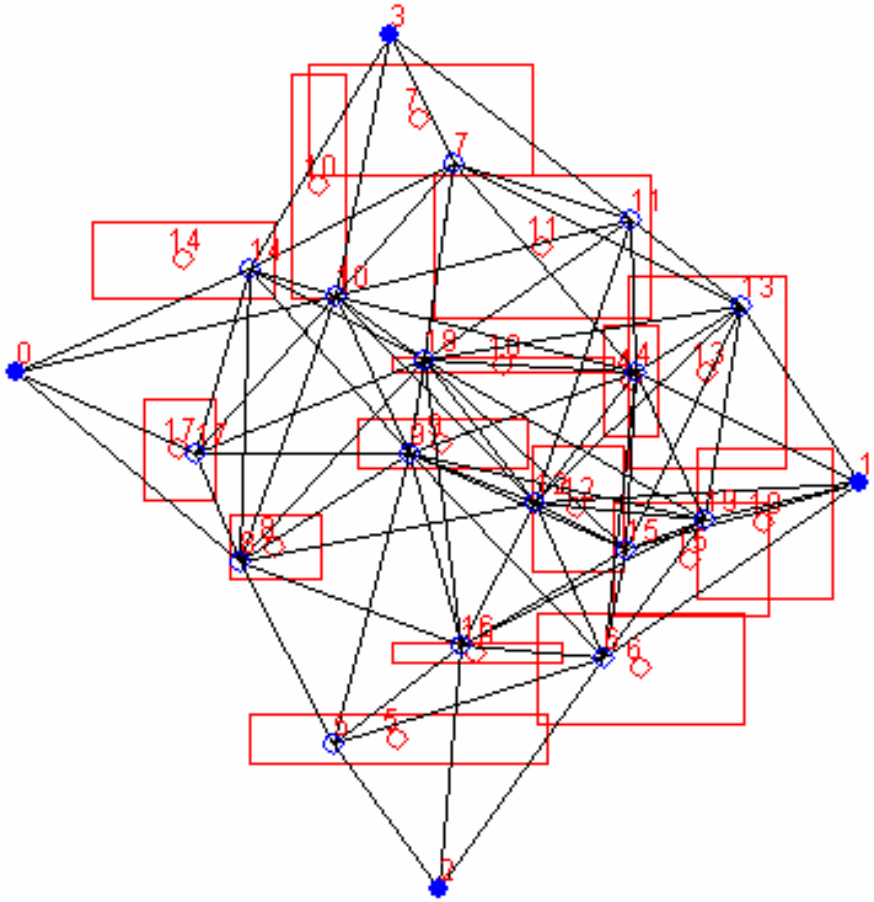
Initial Estimates (Phase 2)

- Use the distance to a beacon as bounds on the x and y coordinates
- Do the same for beacons that are multiple hops away
- Select the most constraining bounds
- **Set the center of the bounding box as the initial estimate**



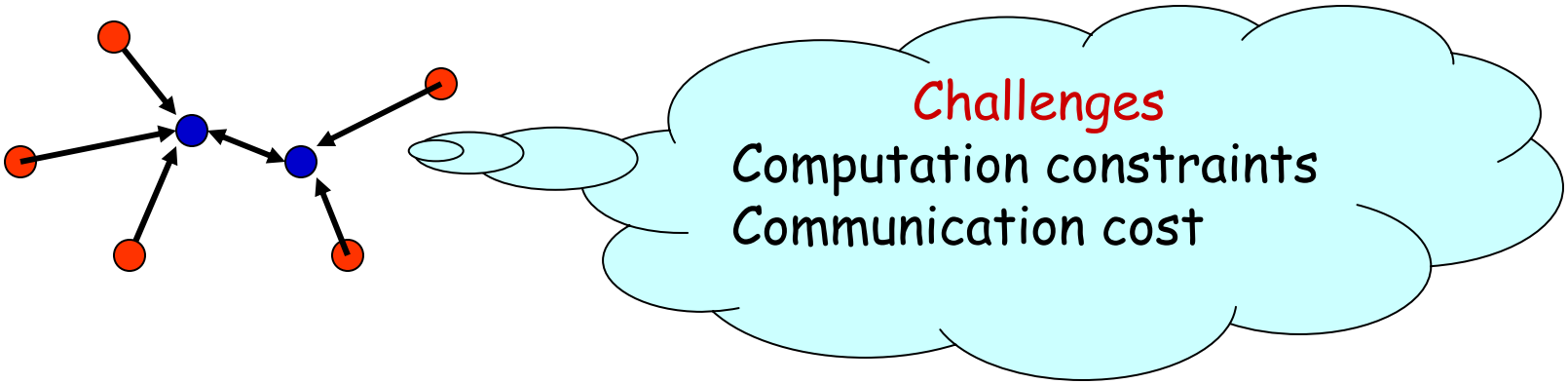
Initial Estimates (Phase 2)

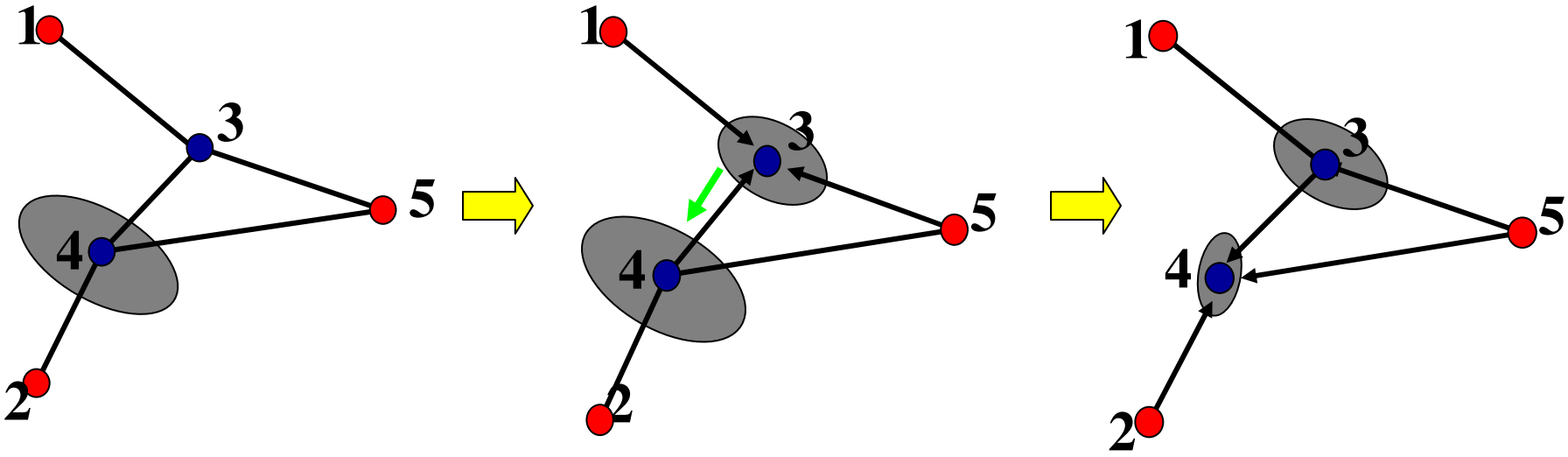
- Initial estimates give rough location information.
- Use Kalman Filter to refine.
  - Start with prior info.
  - Incorporate new measurement info.
  - Improve the current state
  - Details omitted.



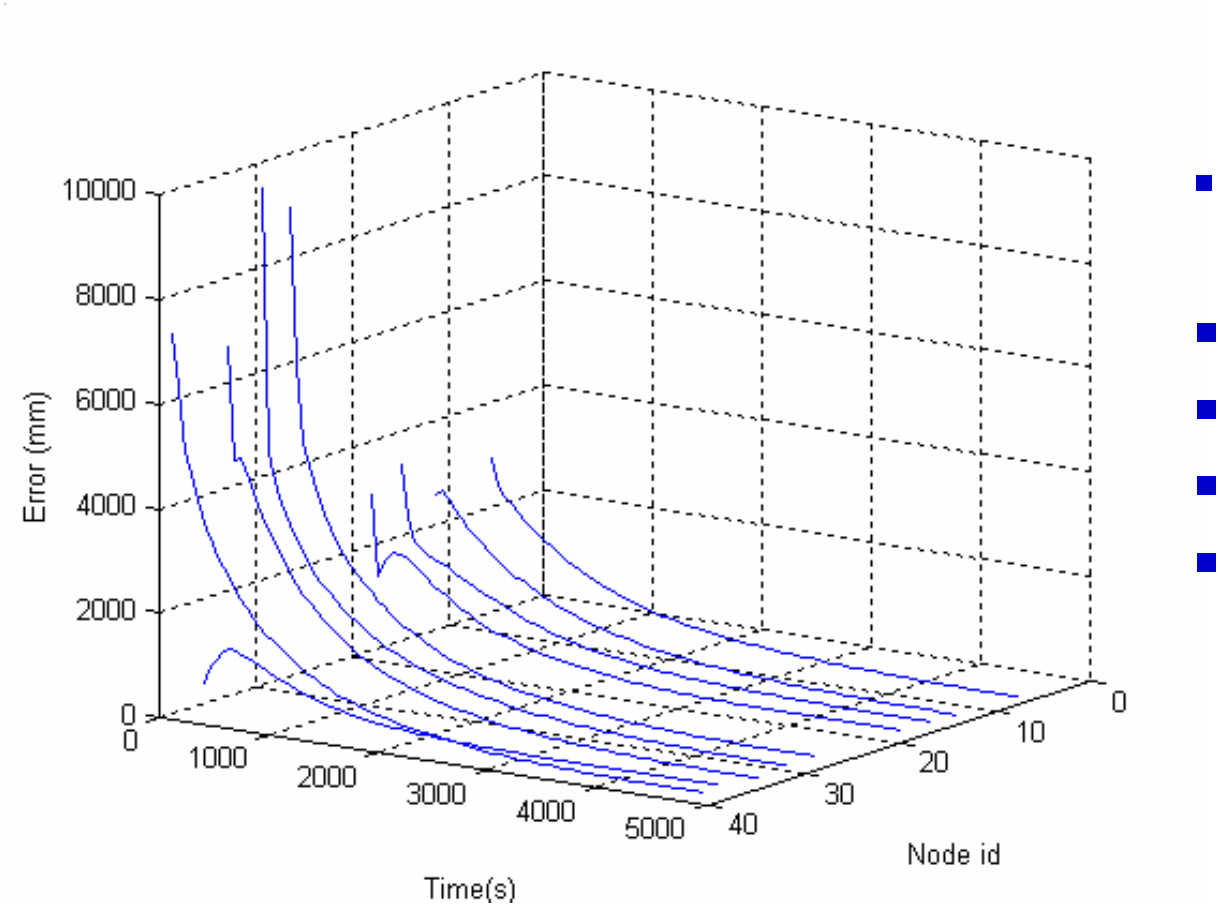
Collaborative Multilateration

Collaborative Multilateration





Satisfy Global Constraints with Local Computation



- From SensorSim simulation
- 40 nodes, 4 beacons
- IEEE 802.11 MAC
- 10Kbps radio
- Average 6 neighbors per node

Multilateration

- Need beacons.
- Iterative multi-lateration.
  - Error accumulates.
  - May get stuck when the density is low.
- Collaborative multi-lateration.
  - Still requires a large number of beacon nodes, especially when the network is sparse.
  - Kalman filter computation is expensive on large networks.

Summary

- Tri-laterations
- Multi-trilaterations.
- Major issue
  - How to deal with noises?
  - How to propagate location information?
- Next class
  - Mass-spring optimization to reduce error.
  - In-correct global layout.