EECS 369: Introduction to Sensor Networks

Winter 2011

Instructors:

Peter Scheuermann and Goce Trajcevski

NORTHWESTERN UNIVERSITY

Part 3: Self-Configuration

3.1. Time Synchronization

3.2. Localization

Self Configuration in Sensor Networks - Motivation

Ad-Hoc Deployment

Rapid Infrastructure Setup

Operate in the presence of obstacles

Self-configuration Challenges

1. Timing synchronization
   - When did an event take place?

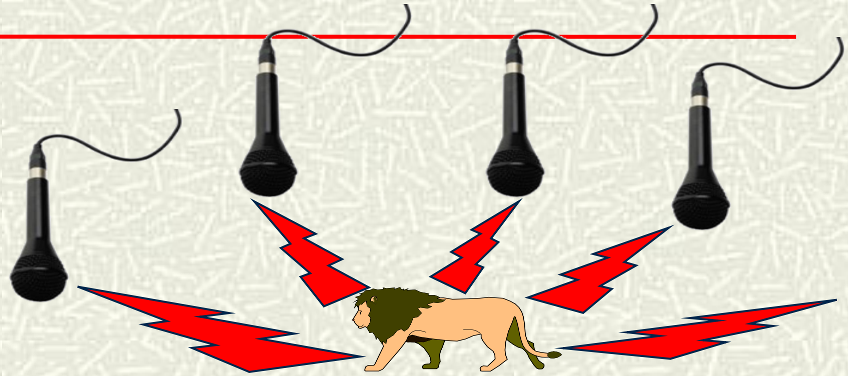
2. Node Localization
   - Where did an event take place?

3. Calibration
   - What is the value of an event?

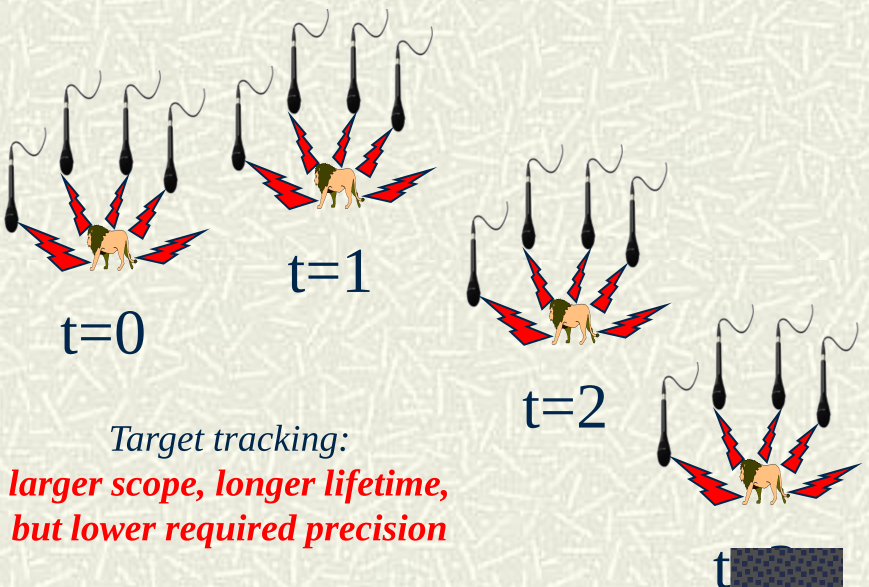
Self-configuration crucial to relate to the physical world!

Importance of Time

- Beam-forming, localization
- Data aggregation & caching
- Security protocols
- MAC layer design
- Adaptive topology management schemes
- Absolute time of occurrence
- Coordinated robotics
- Debugging
- User Interface
- ……



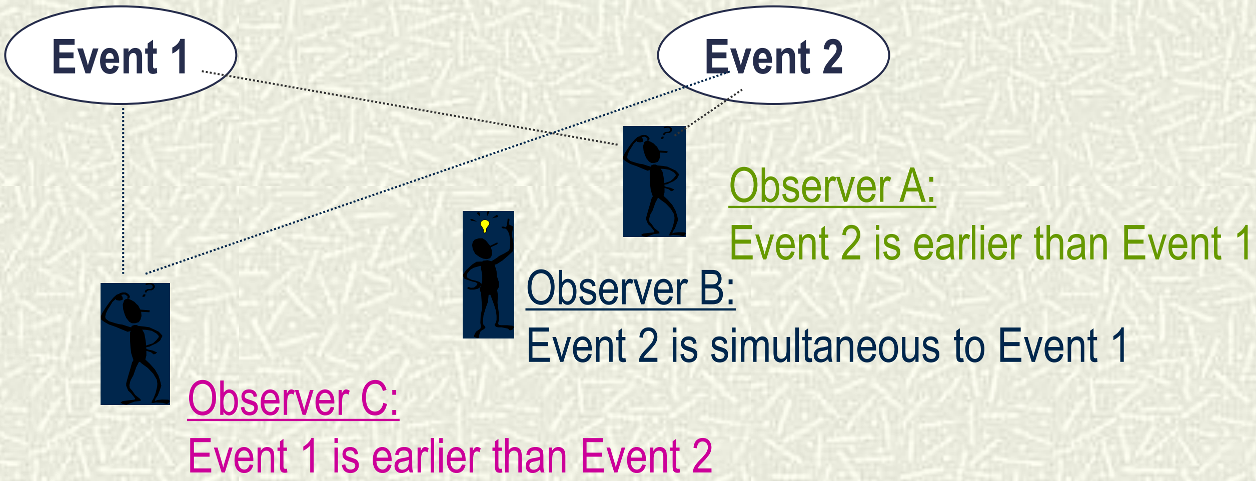
Beam-forming, localization, distributed DSP:
small scope, short lifetime, high precision



Target tracking:
larger scope, longer lifetime,
but lower required precision

Why synchronized time?

The Myth of Simultaneity: “Event 1 and event 2 at same time”



- Ordering of events
- Coordinated actuation
- Data logging
- Absolute time of occurrence
- Performance measurement
- …..

Time Synchronization Quality Metrics

- Maximum Error
- Lifetime
- Scope & Availability
- Efficiency (use of power and time)
- Cost and form factor

…

Time & Clocks

- Clocks: Measure of time!
  - Oscillator and counter

$$C(t) = k\int_{t_0}^{t} \omega(\tau)d\tau + C(t_0)$$

- Synchronized time -> Synchronized clocks

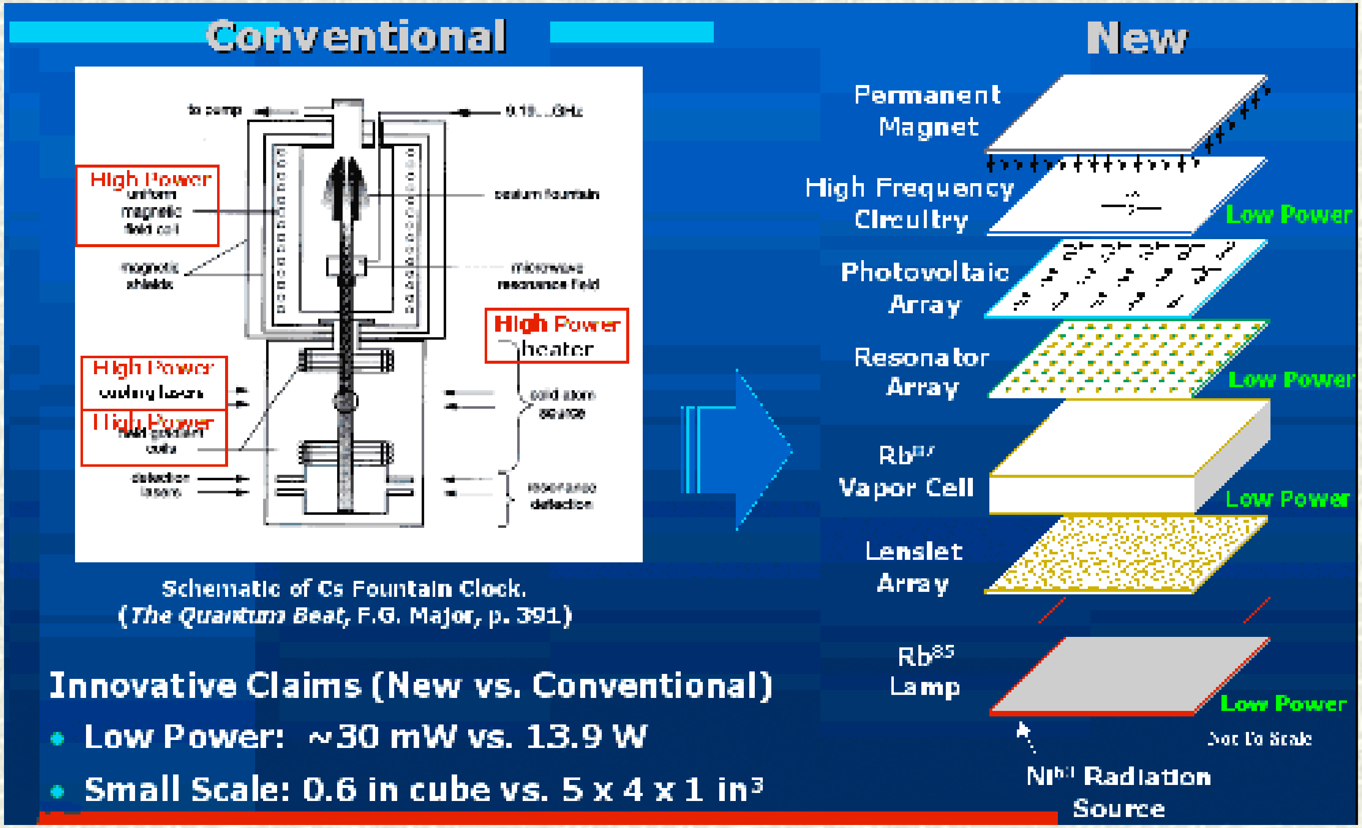
- Errors
  - Clock skew (**offset**): Difference between time on two clocks.
    - Different start times
  - Clock **drift**: Count at different rates.
    - Different frequency of the oscillator.

$$1-\rho \le \frac{dC}{dt} \le 1+\rho$$

Oscillators

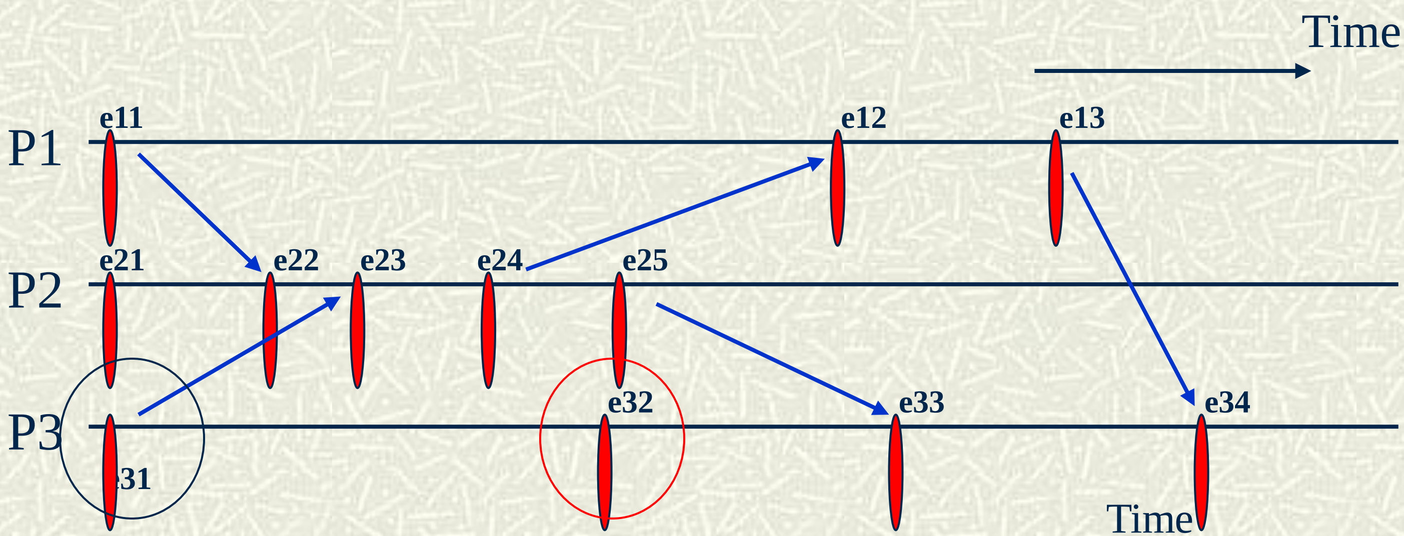
		Drift rate	Cost
Rubidium	Atomic Clocks	$\sim 10^{-15}$ s / day	High
Cesium		$\sim 10^{-12}$ s / day	
Quartz	Desktop	$\sim 10^{-6}$ s / day	
	IPAQs	$\sim 10^{-6}$ s / hr	
	Motes	$\sim 10^{-6}$ s / s	Low

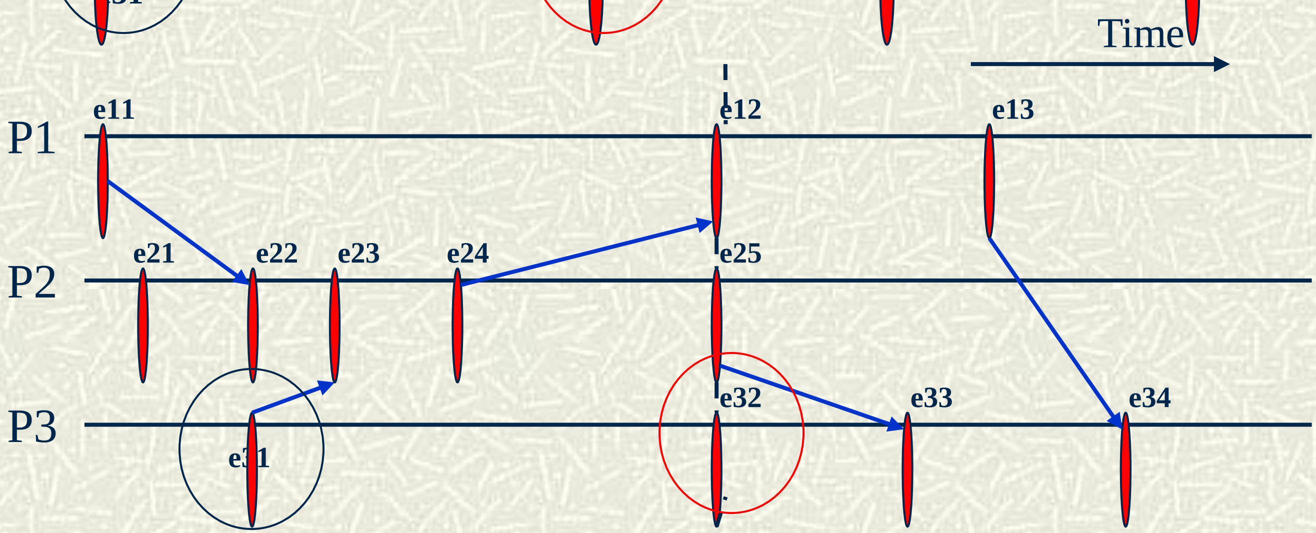
Atomic Clock



CPU power drawn by motes ~ 25 mW
Cost will be the deciding factor!

Logical Clocks: "Time" = Number Assigned to an Event Satisfying Causality





"Time, Clocks, and the Ordering of Events in a Distributed System", Leslie Lamport, Communications of the ACM, July 1978, 21(7)

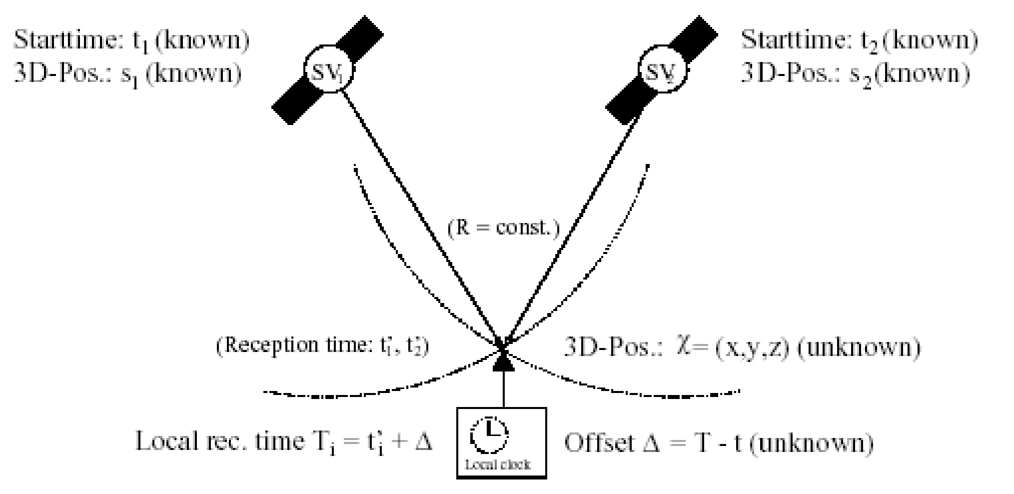
Implementing Logical Clock

- With synchronized ***physical clocks***
  - An event *a* happened before an event *b* if *a* happened at an earlier time than *b*
- ***Without*** physical clocks:
  - Happened before relation "→"
    - If *a* and *b* are events in the same node, and *a* comes before *b*, then $a \rightarrow b$
    - If *a* is the sending of a packet by one node and *b* is the receipt of the same message by another node, then $a \rightarrow b$
    - If $a \rightarrow b$ and $b \rightarrow c$, then $a \rightarrow c$
  - Local clock C_i for each node N_i
    - Assigns a number $C_i(a)$ to any event at a node
  - Each node N_i increments C_i between any two successive events
    - Ensures event ordering ***within*** a node
  - (a) if event a is the sending of a message m by node N_i, then the message m contains a timestamp $T_m = C_i(a)$, and
  - (b) Upon receiving a message m, node N_j sets C_j greater than or equal to its present value and greater than T_m
    - Ensures event ordering across nodes
- Using this method, one can assign a unique timestamp to each event in a distributed system to provide a total ordering of all events
  - But not enough for many applications!

NORTHWESTERN UNIVERSITY

Technologies for Absolute Time Synchronization

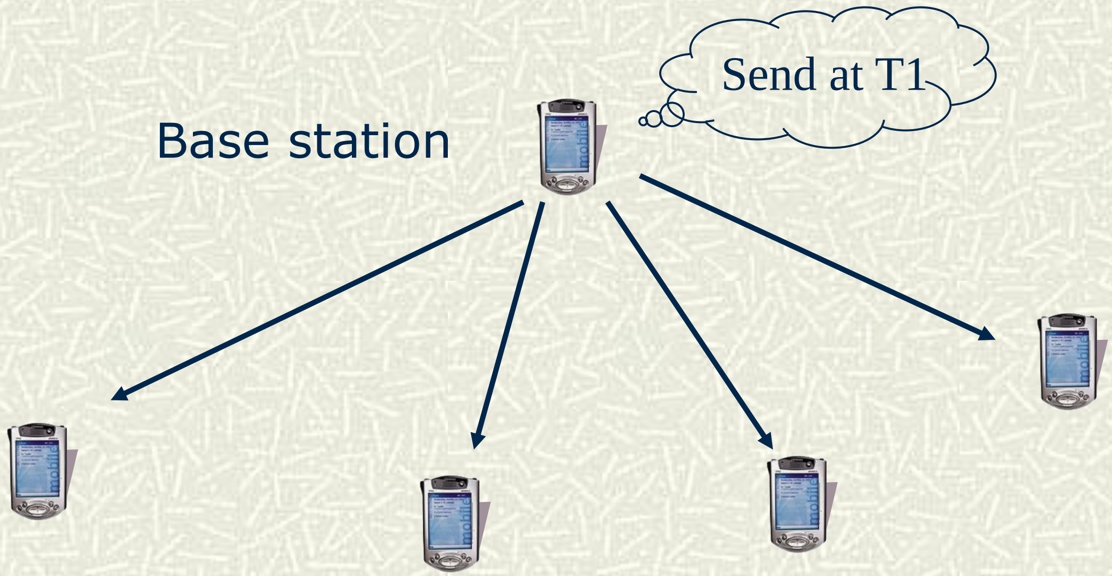
- **GPS**
  - Pros
    - Accuracy ~ 10-100ns (1 PPS signal in GPS)
    - Reliable operation
  - Cons
    - Cannot work indoors, with foliage, obstructions, under water
    - If something goes bad, delay for correcting it can be as large as 30 minutes.
  - Neutral
    - Expensive ~ Cheapest GPS receiver 50 US dollars.
    - Energy hungry!
    - Not all GPS receivers designed for time
- **NIST Radio Time Service: WWVB @ Fort Collins, CO**
  - Continuously broadcasts time and frequency signals at 60 KHz using a 50 KW radiated power transmitter
  - It path delay is removed (e.g. by averaging), WWVB provides uncertainty of less than 100 microseconds relative to UTC
    - Inexpensive one ~ 0.5s
  - Coverage area (signal > 100 microvolt per meter) varies
    - Contracts during day, expands during night



Why not put a GPS receiver at every sensor node?

- Outages:
  - GPS: foliage etc.
  - WWVB: typically available for ~ 20 hours/day
    - Other 4 hours you are stuck with an uncompensated clock oscillator
- Accuracy
  - Inexpensive receivers don’t give good accuracies: intermittent synchronization, serial port delay and jitter
- Listening on a radio is not cheap - energy!!!

802.11 Synchronization



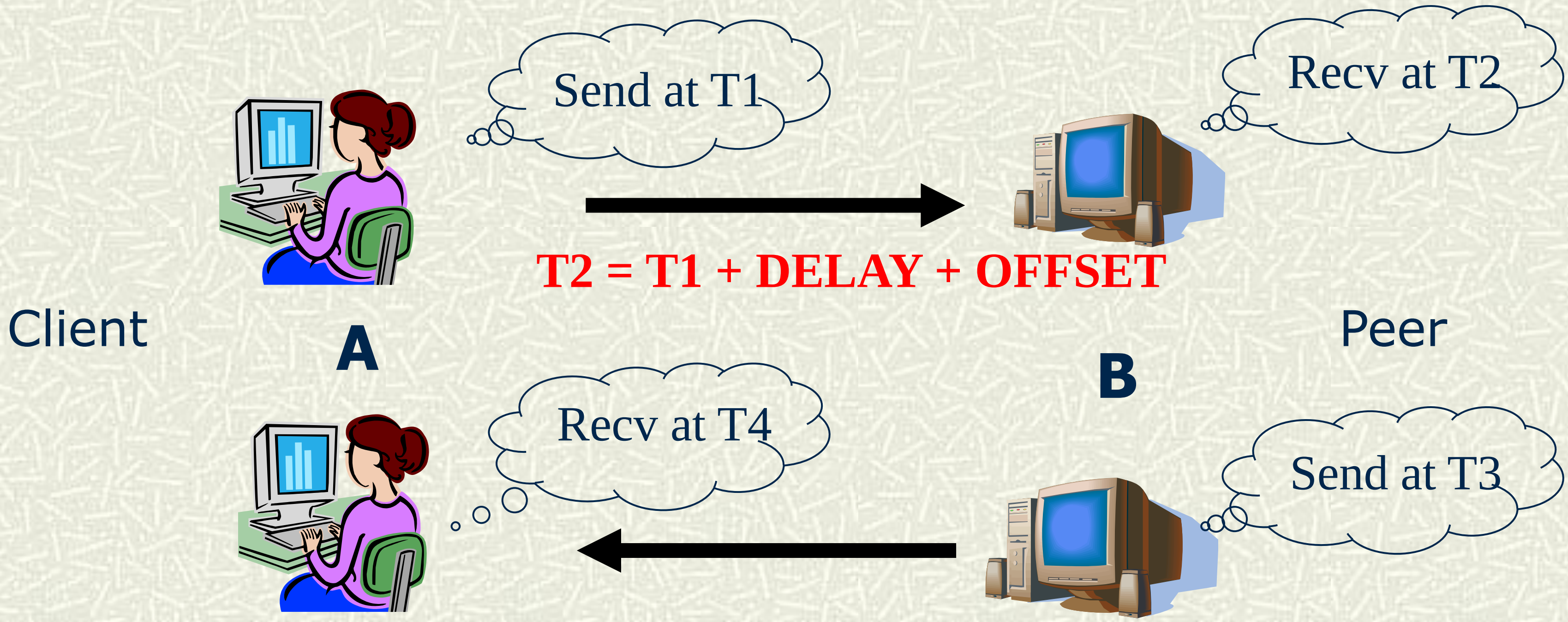
Clients just adopt the timestamp in the beacon packet

Very simple, Provides ms accuracy.

Neglects packet delay and delay jitters

- This approach used by electronic products such as wall clocks, clock radio, wrist watches etc. worldwide to synchronize via WWVB/WWV/WWVH signals
- Can do better by ***compensating*** for propagation delay

NTP: Internet Synchronization



$$T2 = T1 + DELAY + OFFSET$$

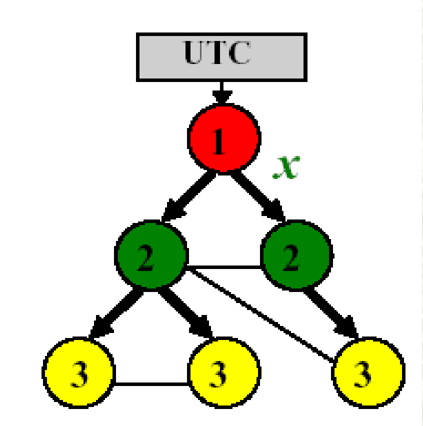
$$T4 = T3 + DELAY - OFFSET$$

$$OFFSET = \{(T2-T1)-(T4-T3)\}/2$$

$$DELAY = \{(T2-T1)+(T4-T3)\}/2$$

NTP details

- Level n synchronizes to level n-1
  - Level 1 synchronizes to UTC via GPS, WWVB etc.
- Multiple synchronization peers -> redundancy and diversity
- Path from clients to a canonical clock is short
- Data filters -> Sliding window
- Intersection and clustering algorithms -> Discard outliers
- Combining algorithm -> Weight samples
- Loop filter and local clock oscillator (LCO) -> implement hybrid phase/frequency-lock feedback loop to minimize jitter



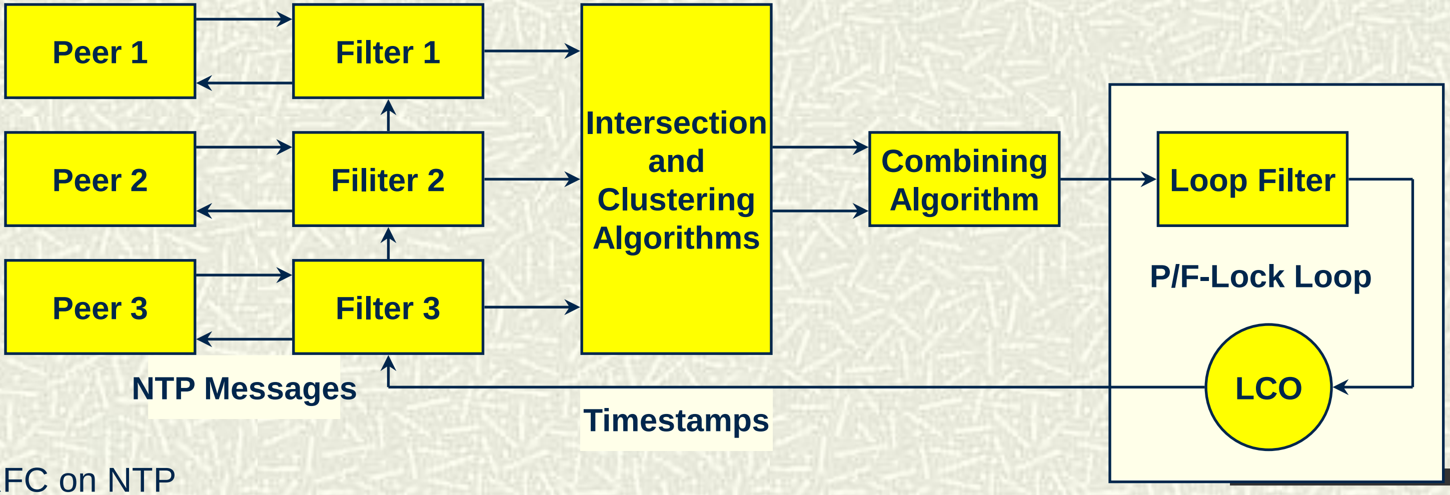


Figure source: RFC on NTP

NORTHWESTERN UNIVERSITY

NTP Evaluation

- Pros
  - Readily available
  - Industry standard
  - Achieves secure and stable sync to ms accuracy
- Cons
  - Designed for ms accuracy only!
  - Not flexible
  - Impact of poor topologies
  - Designed for constant operation in the background at low rates
    - E.g. it took NTP an hour to reduce error to 60 microseconds with maximum polling rate of 16 sec.
- Neutral
  - Not energy friendly!

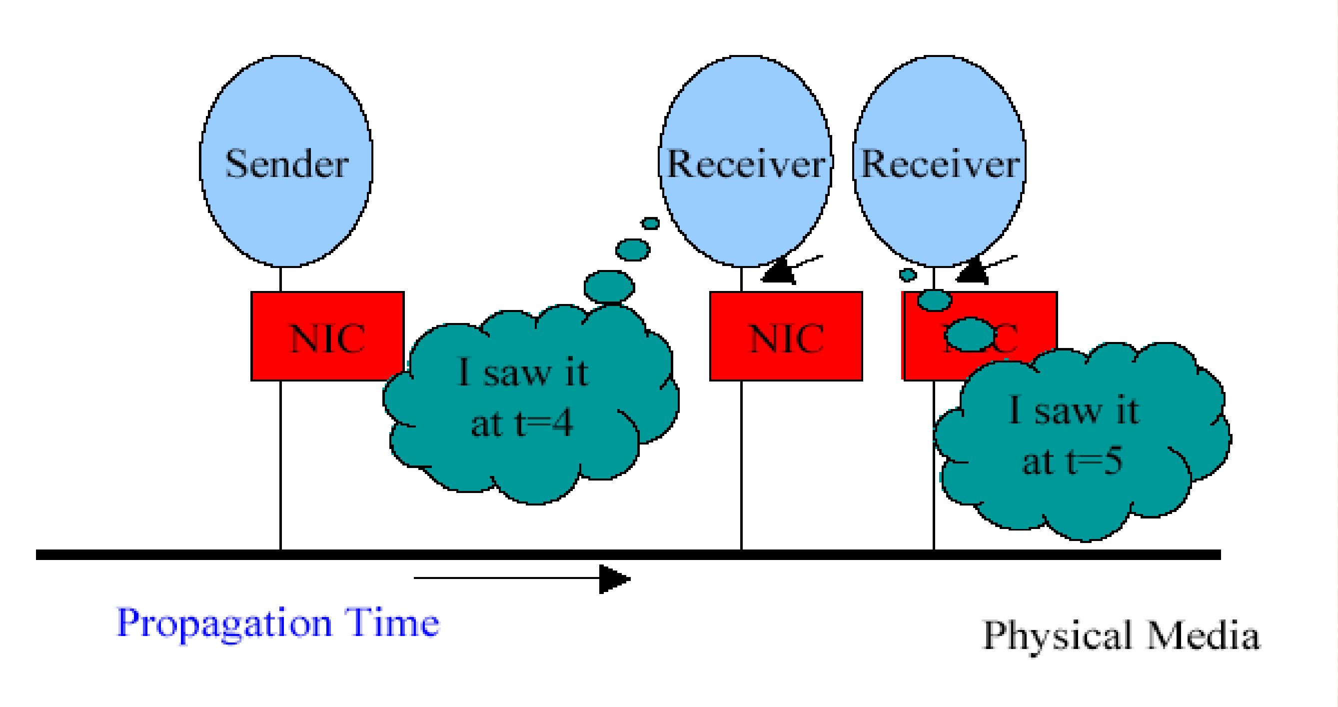
NORTHWESTERN UNIVERSITY

Comparison

Sensor Network Demands	**NTP**	**GPS**
Tunable	One-fit all solution	One-fit all solution
Multi-modal	External synchronization	External synchronization
Energy-efficient	•Listening is free •Using CPU is free •Packet transmissions do not effect lifetime	Receiver energy consumption
µs-ms accuracy	ms accuracy	ns accuracy

Specialized costly hardware
Cannot work indoors

RBS: Synchronize Receivers



Receiver-receiver Synchronization

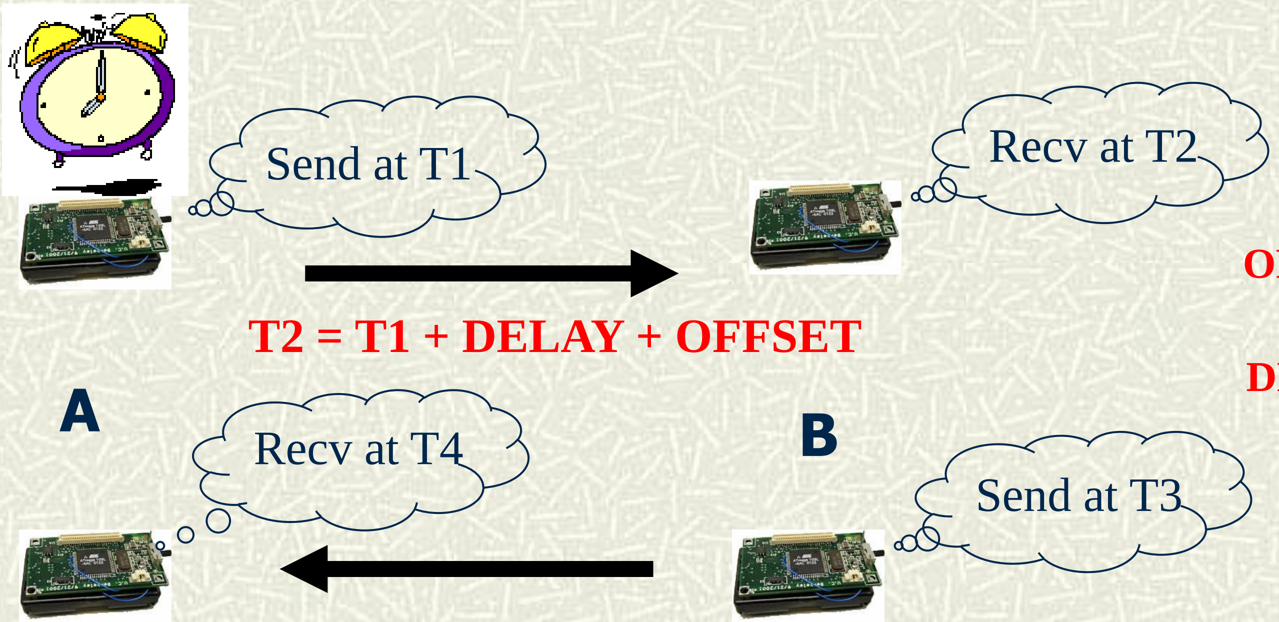
Based on CesiumSpray system by Verissimo and Rodrigues

Phase Offset Estimation

- Simplest case: single pulse, two receivers
  - Xmitter broadcasts reference packet
  - Each receiver records the time that beacon was received according to its *local clock*
  - Receivers exchange observations
  - Sufficient information to form a *local* (relative) timescale
    - However, global timescales are also important
- Extending simple case to many receivers
  - Assumptions
    - Propagation delay is zero
    - No clock skew
    - Receiver *non-determinism* (error) is *Gaussian*
- Sending more messages increases precision
  - Transmitter broadcasts *m* packets
  - Each receiver records time the beacon was observed
  - Receivers exchange observations
  - Receiver *i* computes phase offset to receiver *j* as the *average* of the offsets implied by each pulse received by both nodes
- Result:

$$\forall i \in n, j \in n: \text{Offset}[i, j] = \frac{1}{m} \sum_{k=1}^{m} (T_{j,k} - T_{i,k}).$$

TPSN: Conventional Sender-Receiver Synchronization



T4 = T3 + DELAY- OFFSET

OFFSET = {(T2-T1)-(T4-T3)}/2

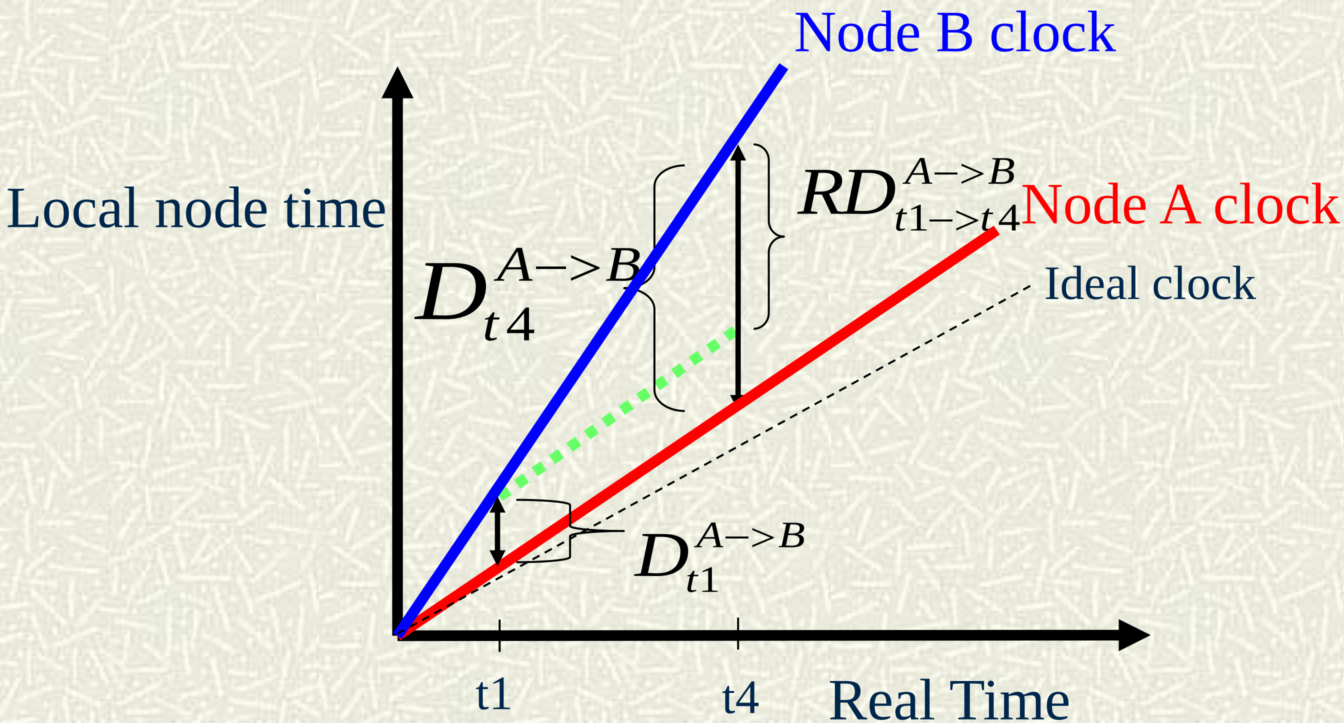
DELAY = {(T2-T1)+(T4-T3)}/2

Based on NTP

The enemy is *non-determinism!*

Also, *asymmetric delays and varying offset*

Variability in Clock Offset

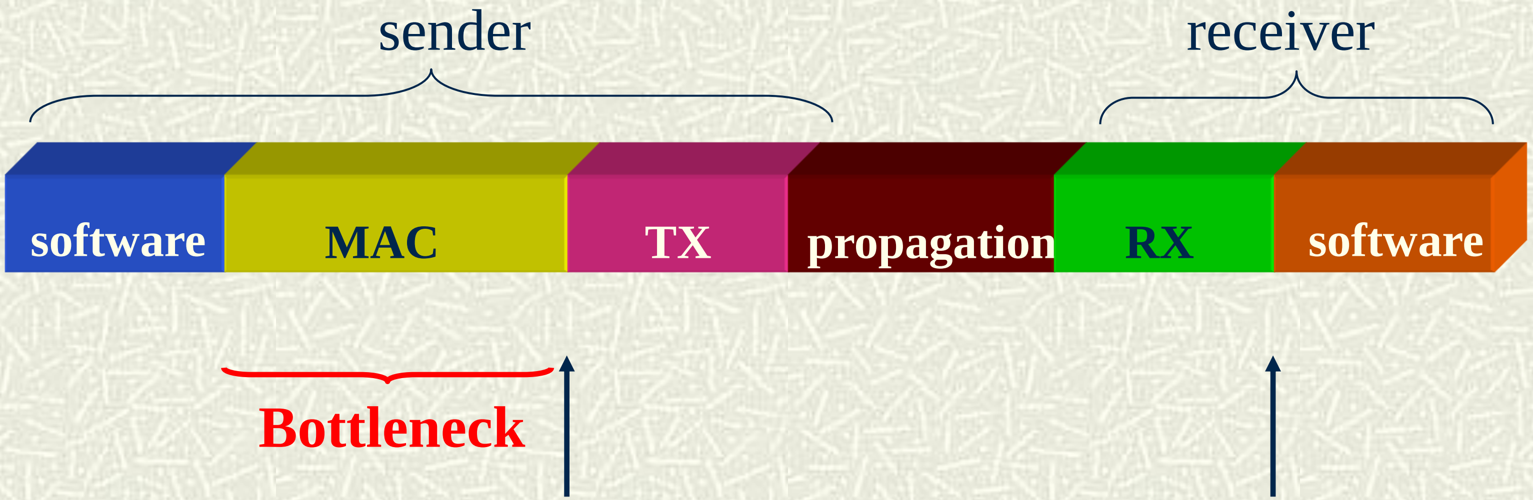


$$Error = \frac{RD_{t1->t4}^{A->B}}{2}$$

Relative offset

Magic behind TPSN

ALL DELAYS ARE VARIABLE !



Use low level time stamping

$$Error = \frac{S^{UC}}{2} + \frac{P^{UC}}{2} + \frac{R^{UC}}{2} + \frac{RD^{A->B}_{t1->t4}}{2}$$

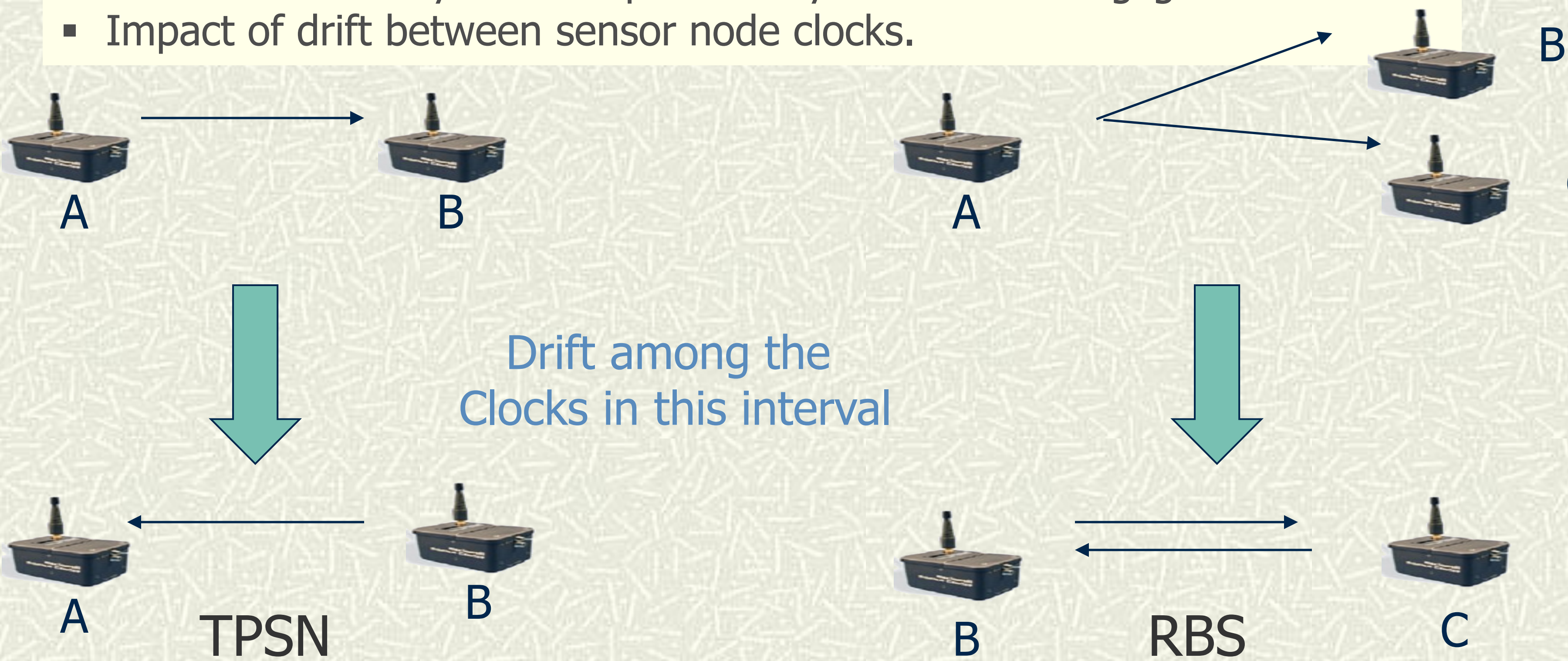
Sender uncertainty

Propagation uncertainty

Receiver uncertainty

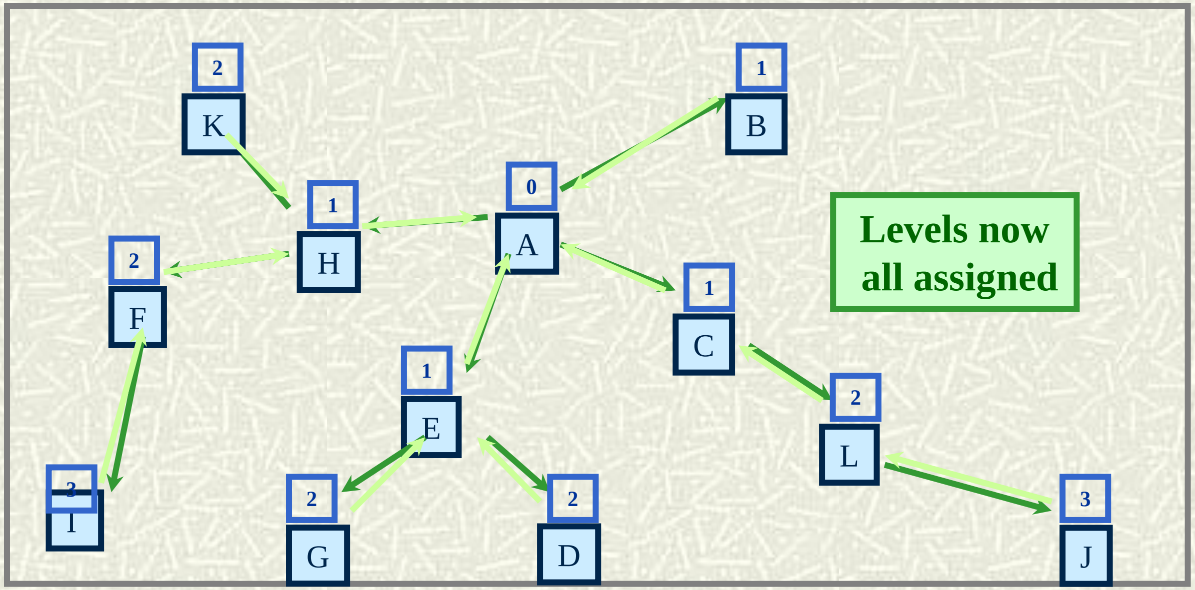
Sources of Error: TPSN vs. RBS

- Most critical is the MAC delay.
- Small variations in software delay and propagation delay.
- Transmission delay and reception delay would have negligible variations
- Impact of drift between sensor node clocks.

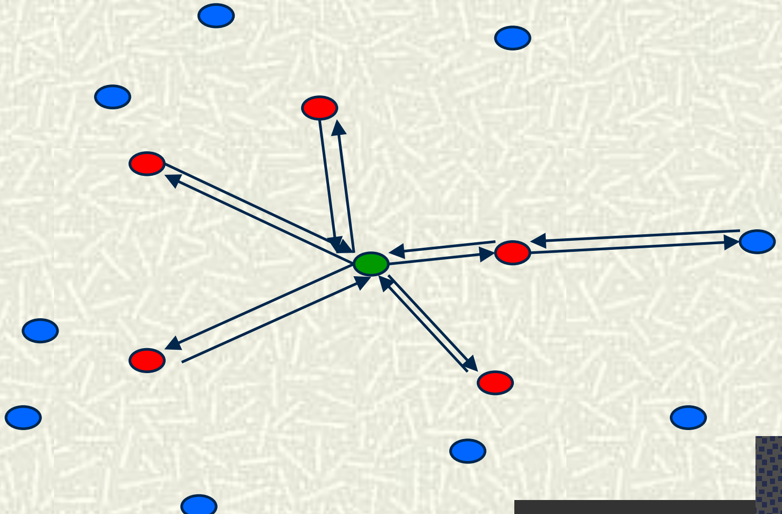


Network-wide Time Synchronization in TPSN

- Level discovery



- Node synchronize to a node belonging to one upper level
  - Use pair-wise synchronization.
  - Needs symmetric links

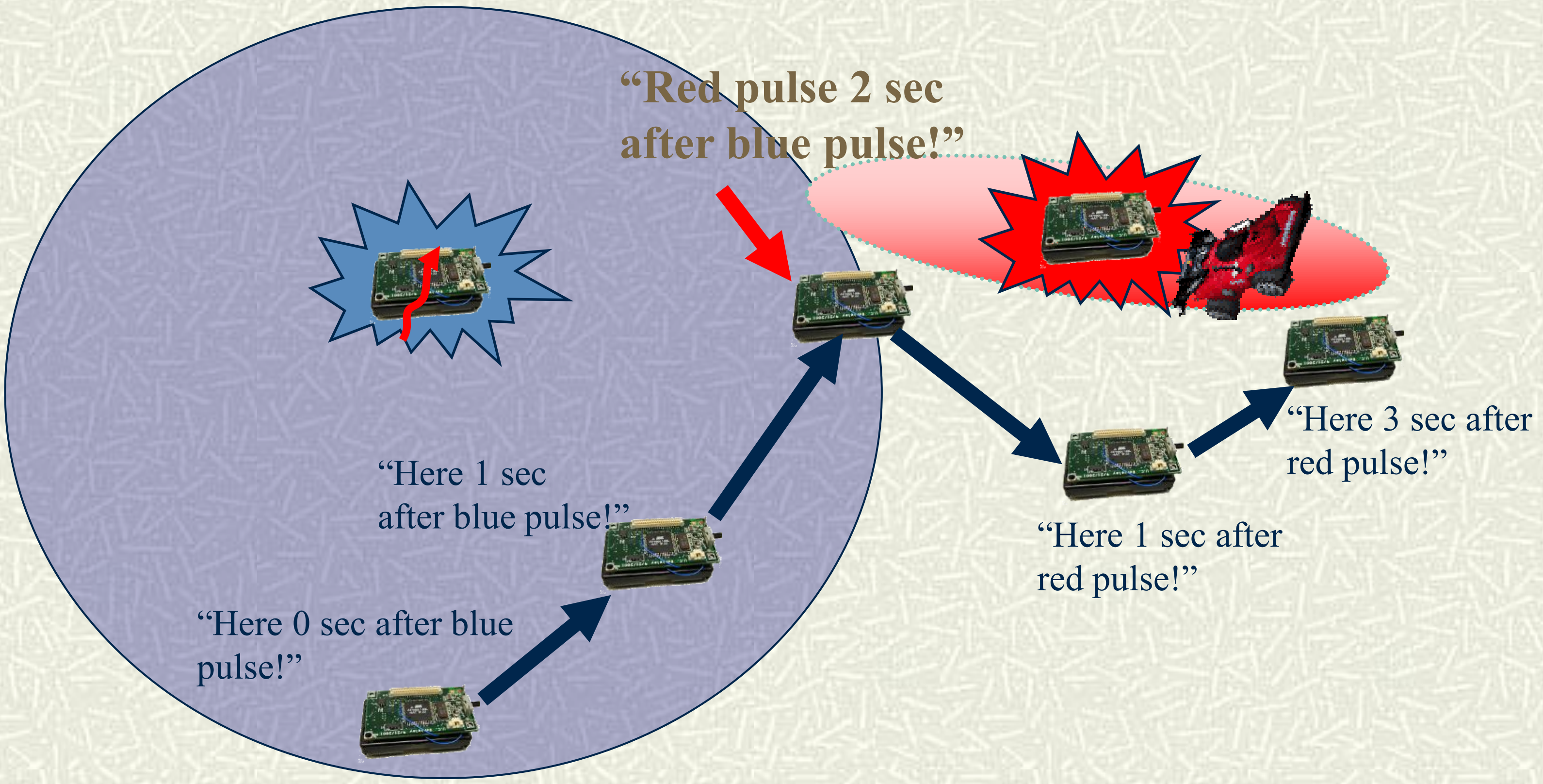


NORTHWESTERN
UNIVERSITY

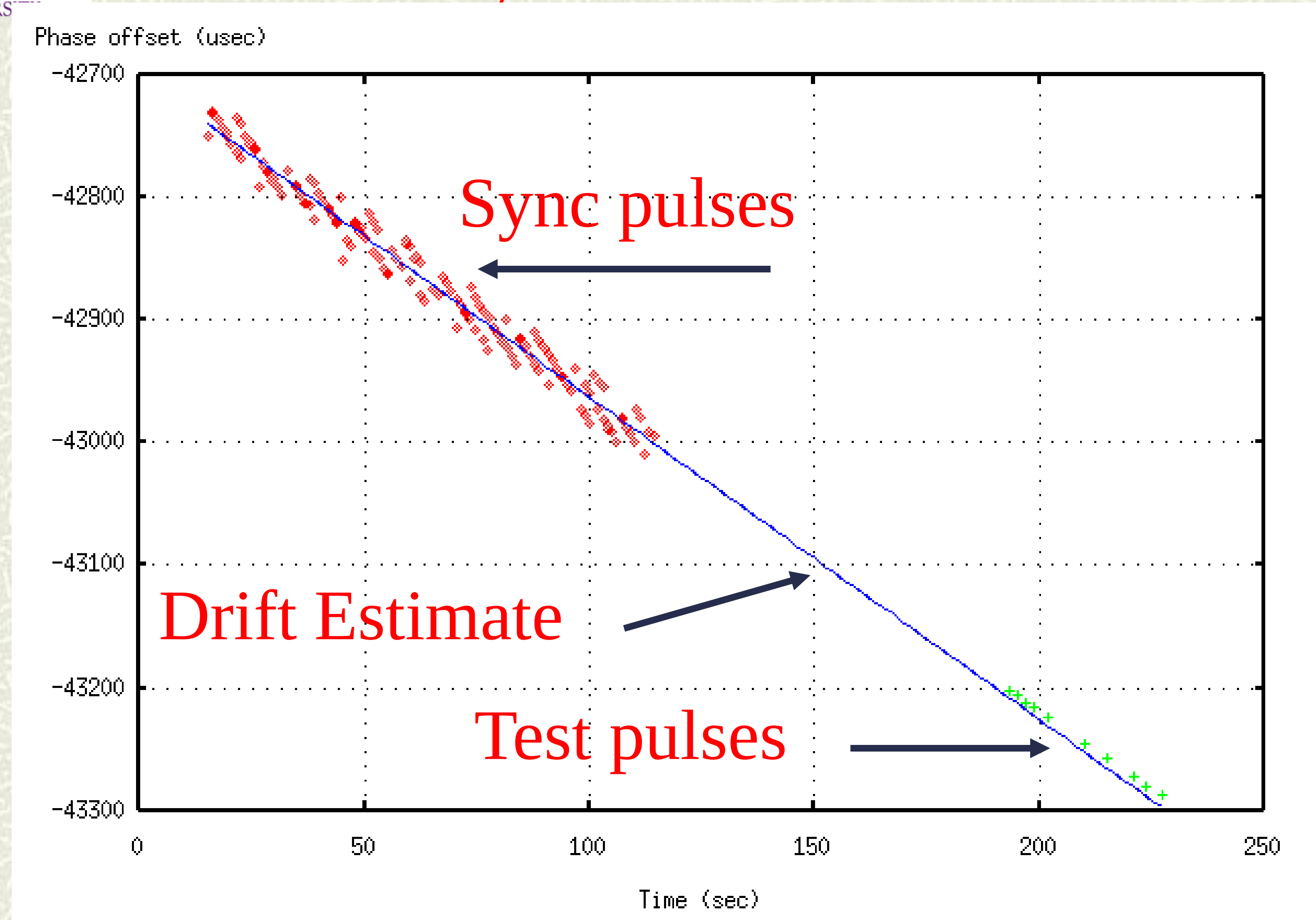
On-demand or Post-Facto Synchronization

- Standard question: what time is it?
  - Requires timing available at highest degree of precision all the time and everywhere
  - Expensive in resources
- New service model: what is the time difference?
  - Need not have global reference
  - Precision depends on purpose (frames, symbols, phase) and can vary throughout the network
  - Allows time-stamping and later resolution of time differences
  - Far lower resource cost
- Approach
  - Clocks start out unsynchronized
  - A set of receivers waits for an interesting event
  - Locally timestamp an event when it happens
  - After the fact, reconcile clocks
- Avoids wasting energy on unneeded sync; it’s easier to predict the past than future

Post-facto Synchronization



Post-facto Synchronization

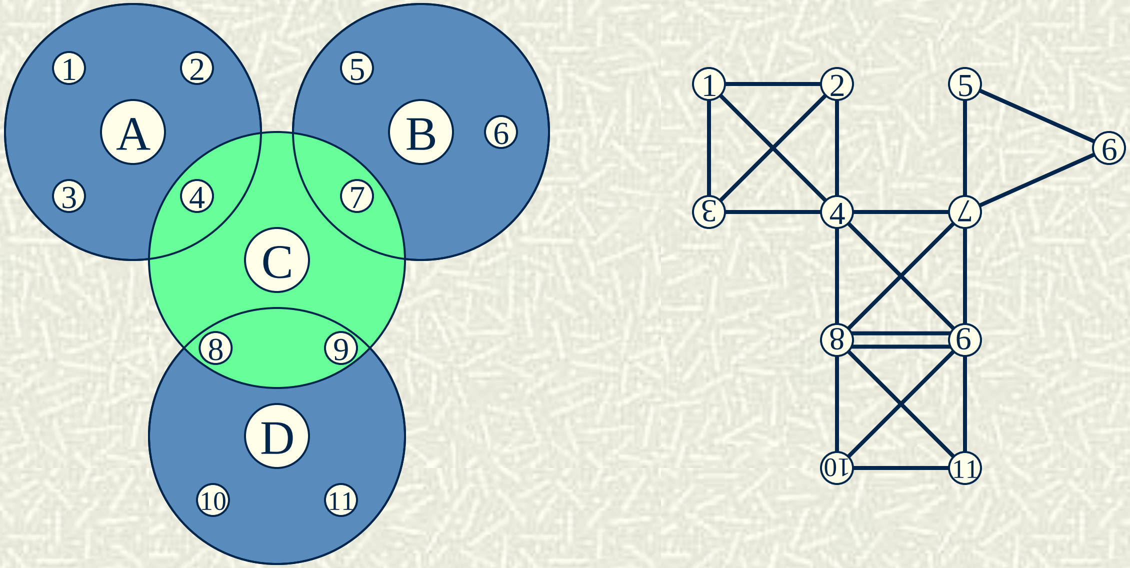


7usec error after 60 seconds of silence

Ref: based on slides by J. Elson

Time Routing

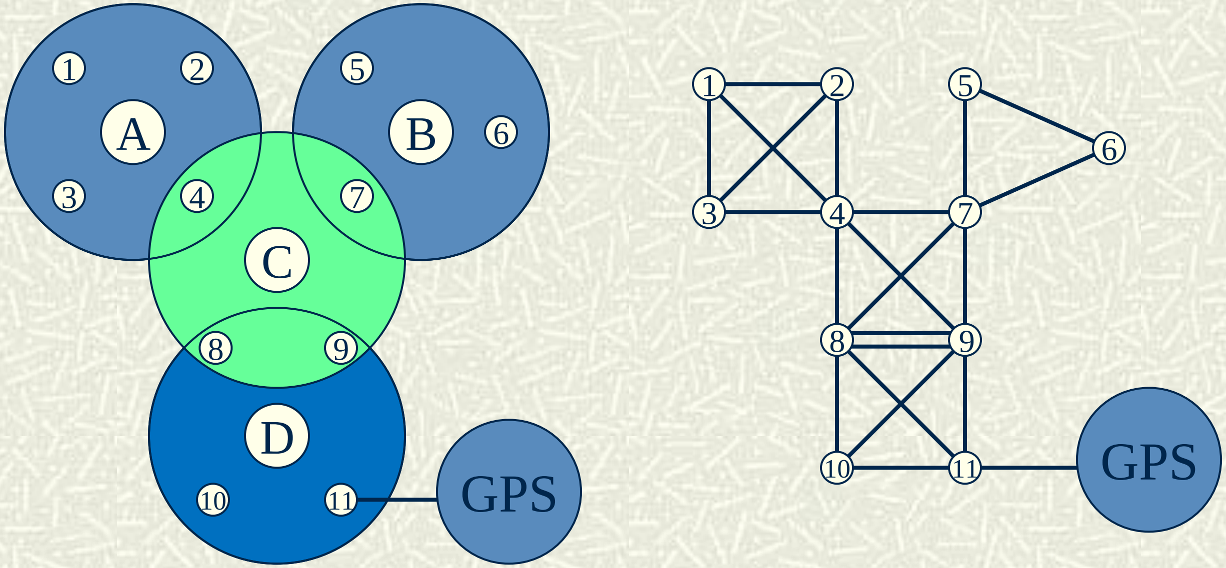
The physical topology can be easily converted to a logical topology; links represent possible clock conversions



Use shortest path search to find a “time route”; Edges can be weighted by error estimates

Synchronizing to External Standards

The multihop algorithm can also be easily used to sync to an external standard such as UTC



E.g. in RBS GPS's PPS generates a series of "fake broadcasts": "received" by node 11's local clock and UTC

Localization - Motivation

Sensor's reading is too hot!!! **WHERE???**

- **Track items (boxes in a warehouse, badges in a building, etc)**
- **Identify items (the thermostat in the corner office)**
- **Not everything needs an IP address**
- **Cost and Physical Environment**
- **Energy Efficiency**

Well, GPS does not work everywhere

- **Smart Systems – devices need to know where they are**
- **Geographic routing & coverage problems**
- **People and asset tracking**

Localization - Challenges

- Physical Layer Measurement Challenges:
  Multipath, shadowing, sensor imperfections, changes in propagation properties and more

- Computational Challenges
  * Many formulations of localization problems:
  (e.g., how to solve the optimization problem, distributed solution)

Plus:

- May not have base stations or beacons for relative positioning
- GPS may not be available
- Sensor nodes may fail
- Low-end sensor nodes

Localization Techniques

1. Electromagnetic Trackers:
   - High accuracy and resolution, but VERY expensive
2. Optical Trackers (Gyroscope):
   - Robust, high accuracy and resolution, expensive and mechanically complex; calibration needed.

3. Radio Position Systems (such as GPS):

Successful in the wide area, but ineffective in buildings, only offer modest location accuracy; cost, size and unavailability.

4. GPS-less Techniques

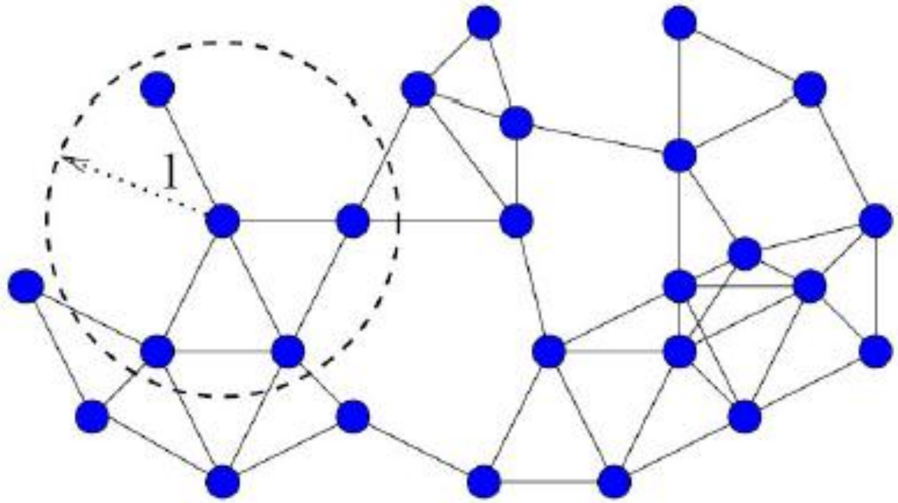
a) Beacon Based Techniques

b) Relative Location Based Techniques

Model of a Sensor Net…

- Typically, when the antenna is omni-directional, the connectivity is modeled via ***Unit disk Graphs***

  – Two nodes have a link if and only if their distance is within 1.



- Use the graph property (connectivity, local measurements) to deduct the locations.

Some Basic Concepts…

- Output: nodes’ location.
  - Global location, e.g., what GPS gives.
  - Relative location.
- Input:
  - Connectivity, hop count.
- Nodes with k hops away are within Euclidean distance k.
- Nodes without a link must be at least distance 1 away.
  - Distance measurement of an incoming link.
  - Angle measurement of an incoming link.
  - Combinations of the above.

Categorization of Localization Approaches

Given distances or angle measurements, find the locations of the sensors.

• **Anchor-based**

– Some nodes know their locations, either by a GPS or as prespecified.

• **Anchor-free**

– Relative location only.

– A harder problem, need to solve the global structure. Nowhere to start.

• **Range-based**

– Use range information (distance estimation).

• **Range-free**

– No distance estimation, use connectivity information such as hop count.

An Example of an Ad-Hoc Approach…

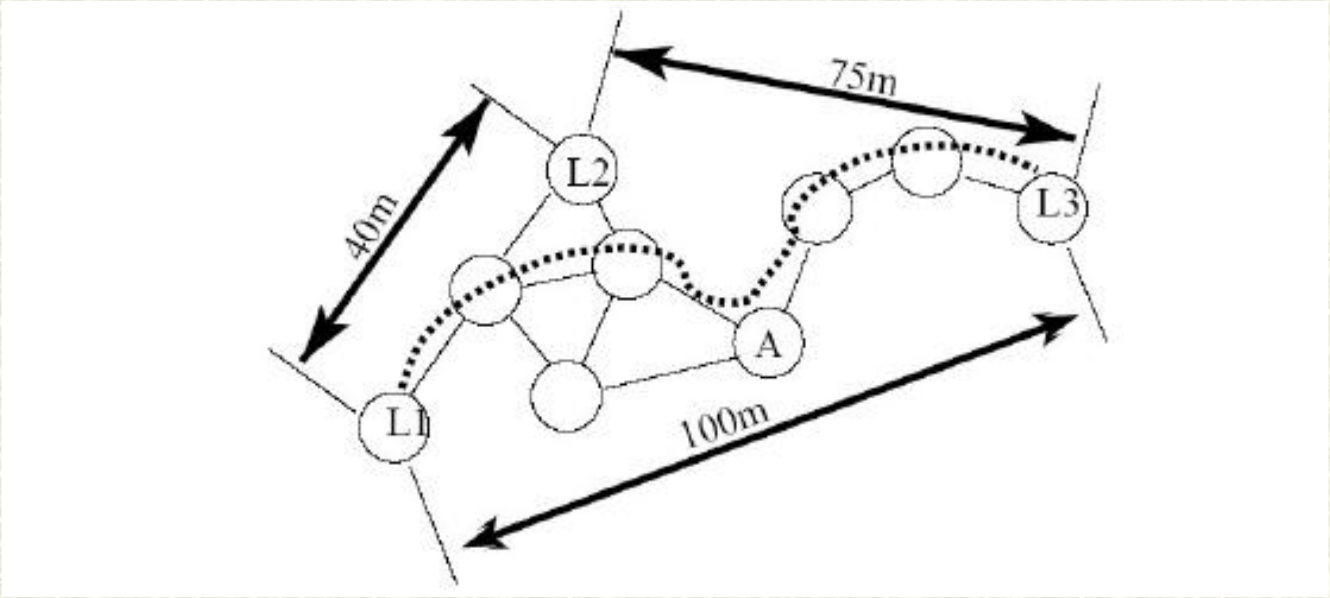
• Ad-hoc positioning (APS) = Estimate range to landmarks using hop count or distance summaries

• APS-basic:

– Count hops between landmarks

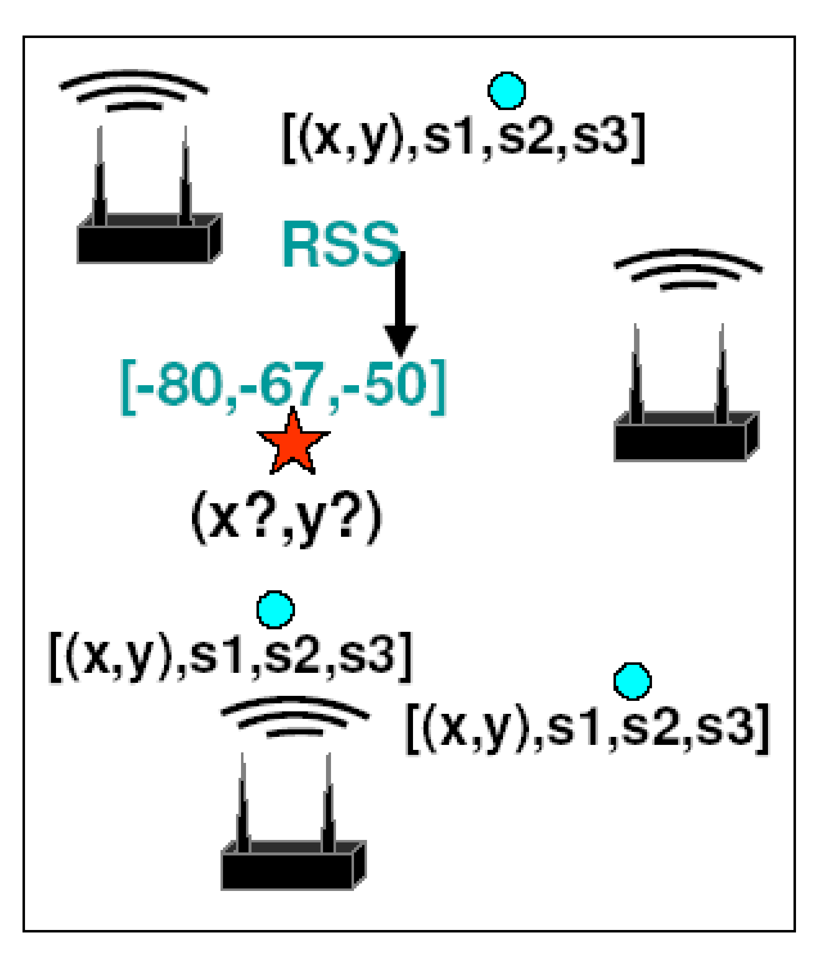
– Find average distance per hop

– Use multi-lateration to compute location



An Example of Fingerprinting Approach…

- Offline phase: collect training data (fingerprints): [(x, y), SS].
  – E.g., the mean Signal Strength to N landmarks.
- Online phase: Match RSS to existing fingerprints probabilistically or by using a distance metric.
- Cons:
  – How to build the map?
- Someone walks around and samples?
- Automatic?
  – Sampling rate?
  – Changes in the scene (people moving around in a building) affect the signal's strength…



NORTHWESTERN UNIVERSITY

GPS Overview

- History
  - U.S. Department of Defense wanted the military to have a super precise form of worldwide positioning
  - After $12B, the result was the GPS system!

- Approach
  - “Man-made stars" as reference points to calculate positions accurate to a matter of meters
  - With advanced forms of GPS you can make measurements to better than a centimeter
  - It's like giving every square meter on the planet a unique address!

GPS Overview

- Constellation of 24 NAVSTAR satellites made by Rockwell
  - Altitude: 10,900 nautical miles
  - Weight: 1900 lbs (in orbit)
  - Size: 17 ft with solar panels extended
  - Orbital Period: 12 hours
  - Orbital Plane: 55 degrees to equitorial plane
  - Planned Lifespan: 7.5 years
  - Current Constellation: 24 Block II production satellites
  - Future Satellites: 21 Block IIrs developed by Martin Marietta

- Ground Stations, aka "Control Segment"
  - Monitor the GPS satellites, checking both their operational health and their exact position in space
  - Five monitor stations
    - Hawaii, Ascension Island, Diego Garcia, Kwajalein, and Colorado Springs.

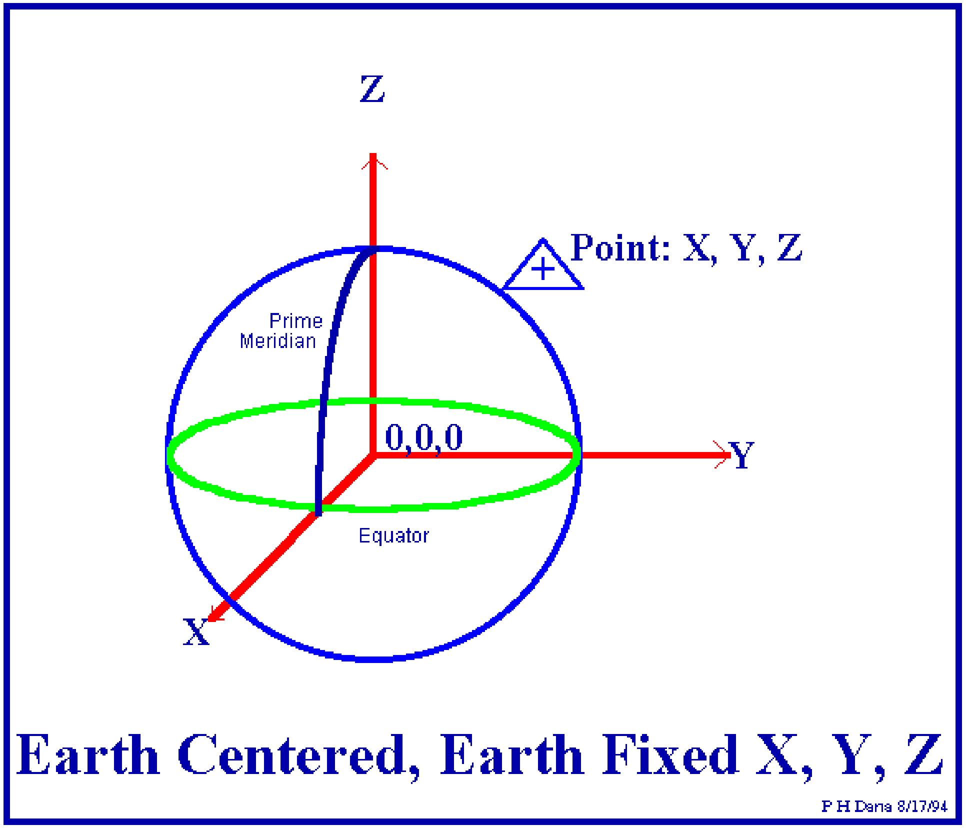
GPS – Basic Operation Principles

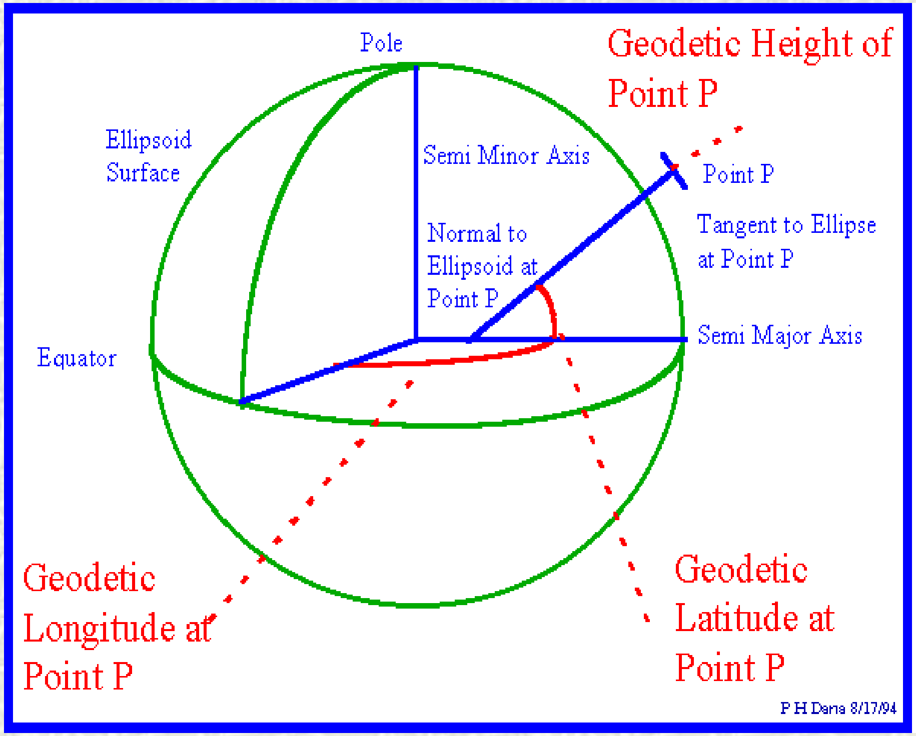
1. The basis of GPS is "trilateration" from satellites. (popularly but ***wrongly*** called "triangulation")
2. To "trilaterate," a GPS receiver measures distance using the travel time of radio signals.
3. To measure travel time, GPS needs very accurate timing which it achieves with some tricks.

4. Along with distance, you need to know exactly where the satellites are in space. High orbits and careful monitoring are the secret.

5. Finally you must correct for any delays the signal experiences as it travels through the atmosphere.

Coordinates





Geodetic Coordinates (latitude, longitude, height)

NORTHWESTERN UNIVERSITY

Trilateration

- GPS receiver measures distances from satellites
- Distance from satellite #1 = 11000 miles
  - We must be on the surface of a sphere of radius 11000 miles, centered at satellite #1
- Distance from satellite #2 = 12000 miles
  - We are also on the surface of a sphere of radius 12000 miles, centered at satellite #2,

  i.e., on the circle where the two spheres intersect

- Distance from satellite #3 = 13000 miles
  - In addition, we are also on the surface of a sphere of radius 13000 miles, centered at satellite #3

  i.e., on the two points where this sphere and the circle intersect

  Could use a fourth measurement, but usually one of the points is impossible (far from Earth, or moving with high velocity) and can be rejected but fourth measurement useful for another reason!

Measuring Distances from Satellites

- By timing how long it takes for a signal sent from the satellite to arrive at the receiver
  - We already know the speed of light
- Timing problem is tricky
  - Smallest distance - 0.06 seconds
  - Need some really precise clocks

Thousandth of a second error → 200 miles of error

- On satellite side, atomic clocks provide almost perfectly stable and accurate timing
- What about on the receiver side?
  - Atomic clocks too expensive!

- OK, but even assuming precise clocks, how do we measure travel times?

Measuring Travel Times from Satellites

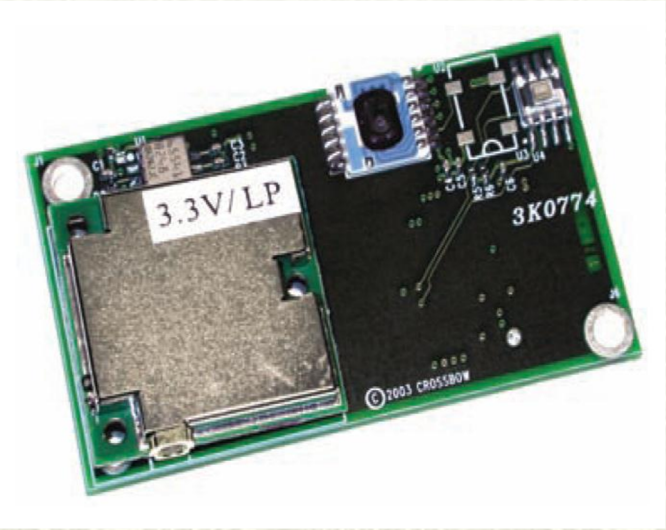
- Each satellite transmits a unique pseudo-random code, a copy of which is created in real time in the user-set receiver by the internal electronics
- The receiver then gradually time-shifts its internal code until it corresponds to the received code--an event called lock-on.
- Once locked on to a satellite, the receiver can determine the exact timing of the received signal in reference to its own internal clock

- If receiver clock was perfectly synchronized, three satellites would be enough
- In real GPS receivers, the internal clock is not quite accurate enough
- The clock bias error can be determined by locking on to ***four*** satellites, and solving for X, Y, and Z coordinates, and the clock bias error

Extra Satellite Measurement to Eliminate Clock Errors

- Three perfect measurements can locate a point in 3D
- Four *imperfect* measurements can do the same thing
  - If there is error in receiver clock, the fourth measurement will not intersect with the first three
- Receiver looks for a single correction factor
- The correction factor can then be applied to all measurements from then on.
  - From then on its clock is synced to universal time.
  - This correction process would have to be repeated constantly to make sure the receiver's clocks stay synched
- ***=> At least four*** channels are required for four simultaneous measurements

GPS in WSNs (oh, well…)



- Xbow MTS420CA: Environmental monitoring sensor board for a mote with "regular" capabilities of Mica2 and MicaZ
  - Tracking channels: 12
  - Position accuracy: 10 m
- However, the ***price*** of the GPS-ability is still very expensive
  - MicaZ node: few 10s of $
  - MTS420CA: few 100s of $
- Also:
  - GPS does NOT work indoors
  - Accuracy (10m) may not be enough for dense WSNs
  - GPS-less techniques are required

GPS-less Techniques

Use DISTANCE or ANGLE measurements from a set of fixed reference points and apply

***MULTI-LATERATION* or *TRIANGULATION* techniques.**

Basic approaches:

a. Received Signal Strength (RSS)

b. Time of Arrival (TOA)

c. Time Difference of Arrival (TDOA)

d. Angle of Arrival (AOA)

Received Signal Strength (RSS)

IDEA:

Use some readily-available info to estimate the distance between a transmitter and a receiver:

a. The Power of the Received Signal

b. Knowledge of Transmitter Power

c. Path Loss Model

Crux:

Each measurement gives a circle on which the sensor must lie…

Note: RSS method may be unreliable/inaccurate due to:

a. Multi-path effects

b. Shadowing, scattering, and other impairments

c. Non line-of-sight conditions

Time of Arrival (ToA)

BASIC IDEA:

Estimates the relative distance to a beacon by applying the measured propagation time to a distance formula (modeling the reality)...

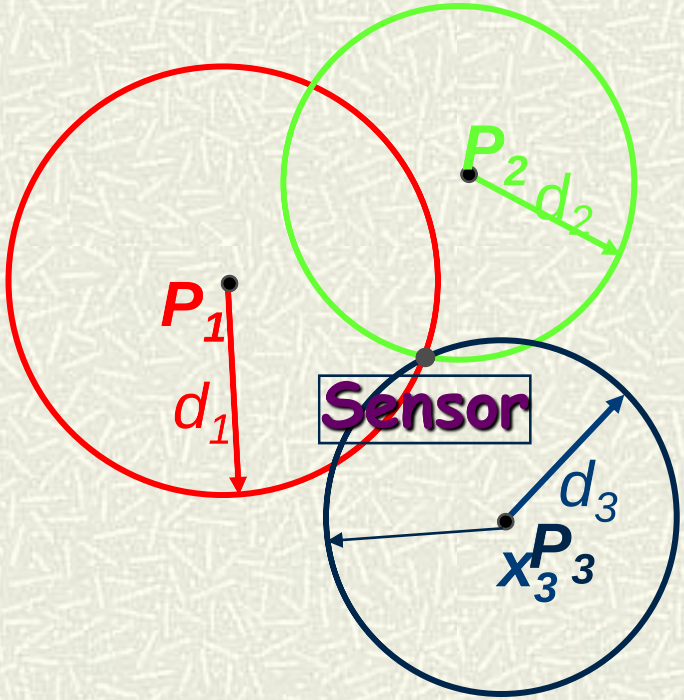
$$\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} + \varepsilon_i(x_0, y_0, s) = st_i$$

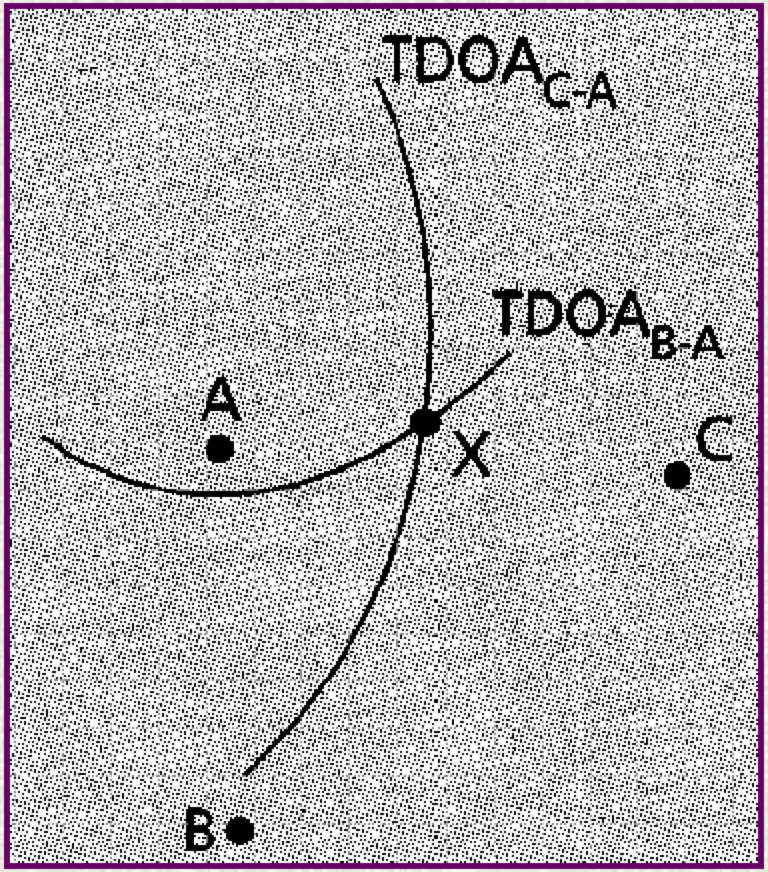
Active: Receiver sends a signal that is bounced back so that the receiver know the round-trip time

Passive: Receiver and transmitter are separate Time of signal transmission needs to be known

Drawback: due to fast propagation speed of wireless signals where a small error in time measurement can result in large distance estimate errors.

Location estimate…





TDoA (hyperbola, actually…)

RSS or ToA

NOTE: strictly speaking two "circles" needed…

Time or Time Difference of Arrival (ToA/TDoA)

- Time synchronized beacons and nodes that send periodic announcement also enable the node to calculate an estimate of its distance from other nodes.

- This can be achieved by calculating the time difference of the expected and actual arrival of a signal, taking into account possible interference and propagation delay.

NORTHWESTERN UNIVERSITY

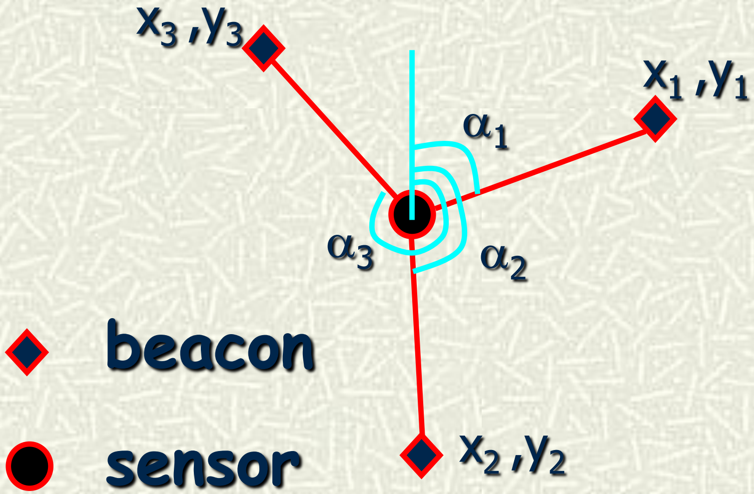
Angle of Arrival (AoA)

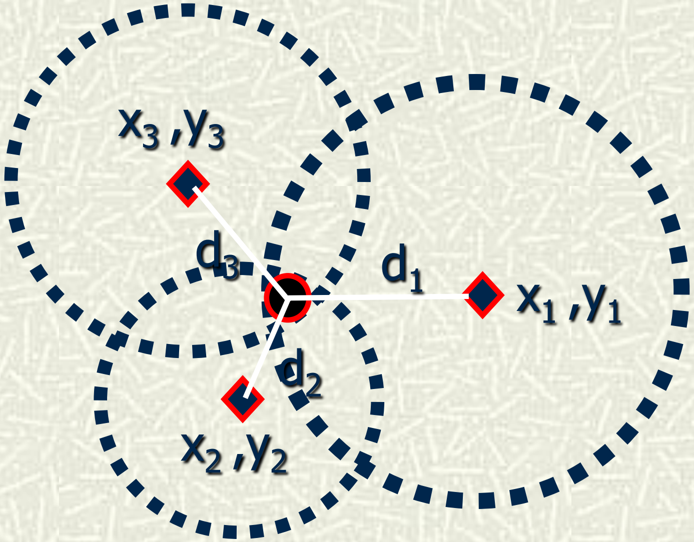
Special antenna configurations are used to estimate the angle of arrival of the received signal from a beacon node.

Angle of arrival method may also be unreliable and inaccurate due to:

a. Multi-path effects,
b. Shadowing, scattering, and other impairments,
c. Non line of sight conditions.

Trilateration and Triangulation

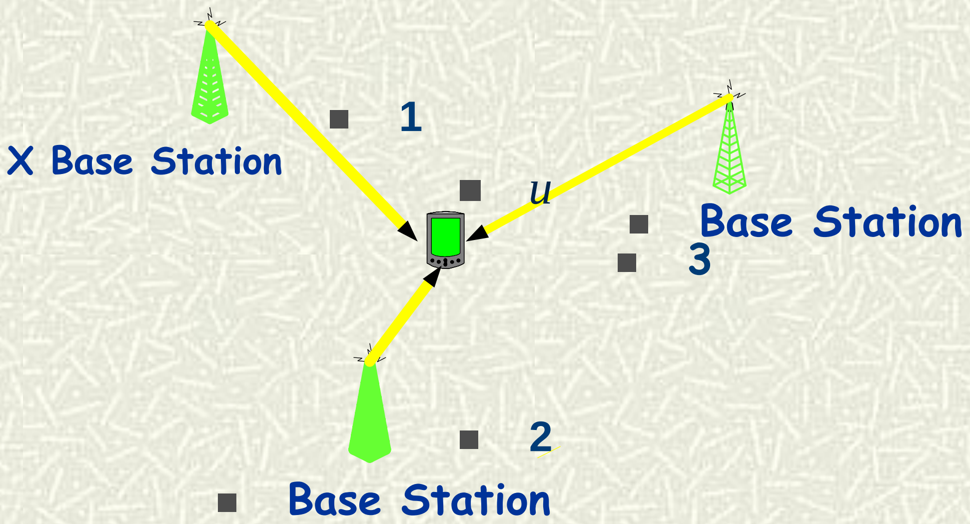




Three or more beacon location and their distance to the node location are known.

Three or more beacon location and
their direction according to the node location are known.

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
- Recall: Distance measurements are noisy!

Mathematical Formulation of the Problem

- 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

Minimize the Mean-Square Error (MMSE)

- 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.

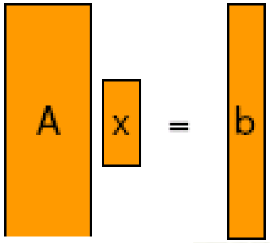
Solution…

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

$$Ax = b$$

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

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



The linearized equations in matrix form become

$$Ax = b$$

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

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

NOTE: even here, some "conditions" may mess it up, e.g., collinear beacons

Q: what is the signal is acoustic???

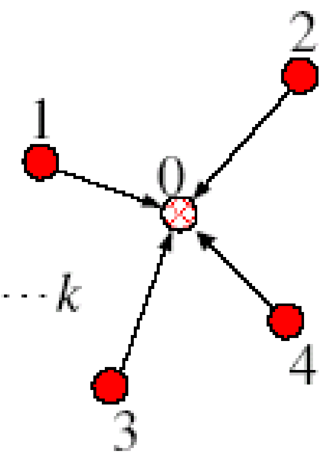
Must take the speed into consideration (delay of arrival…)

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



$i = 1, 2 \cdots k$

This can be linearized to the form

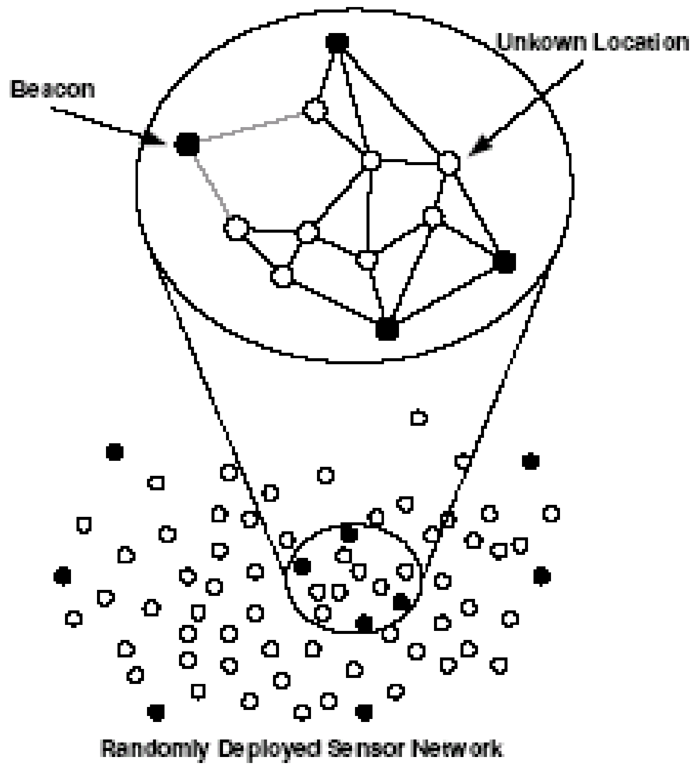
$$Ax = b$$

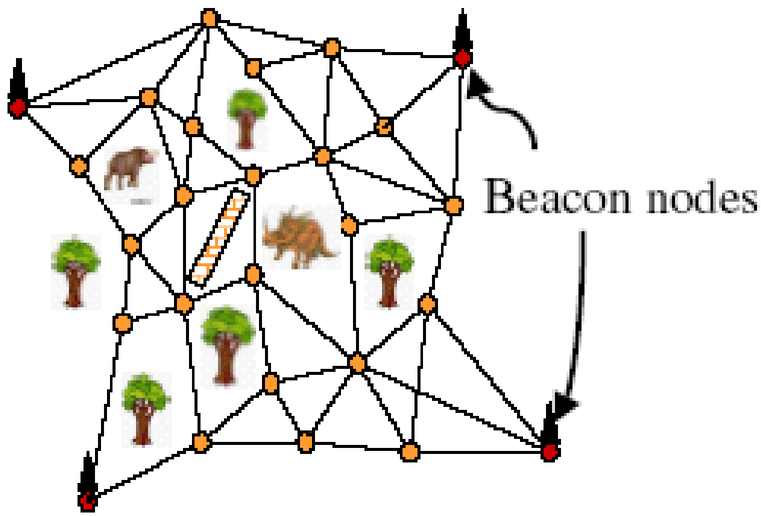
where

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

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

Generalization





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

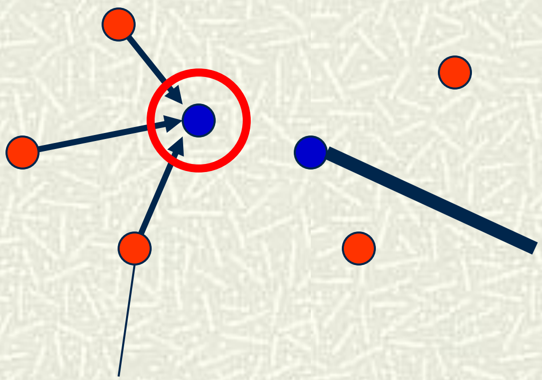
NOTE: better if beacons are mostly around the perimeter…

NORTHWESTERN UNIVERSITY

Solving over Multiple Hops

-A node with 3 beacon neighbors resolves its location and becomes a beacon

AKA Iterative Multilateration



Unknown node (unknown position)

Beacon node (known position)

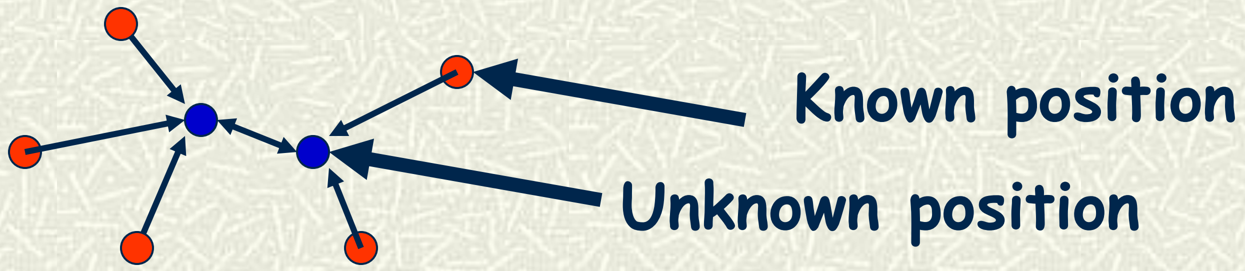
-Must-Connectivity (i.e., each node needs at least 3 neighbors)

Collaborative Multilateration

- Saw example:
  - First, the nodes with 3 beacon-neighbors estimate their location(s)
  - Subsequently, those nodes are used as beacon nodes
- Possible problem
  - Too few beacons
  - Too hard to locate the nodes with beacon neighbors (to continue the “iterations”)
- Solution:
  - Use as much information as possible in initial estimation
  - Proceed with “refining”

Collaborative Multilateration

- All available measurements are used as constraints



- Solve for the positions of multiple unknowns simultaneously
- Catch: This is a non-linear optimization problem!

NORTHWESTERN UNIVERSITY

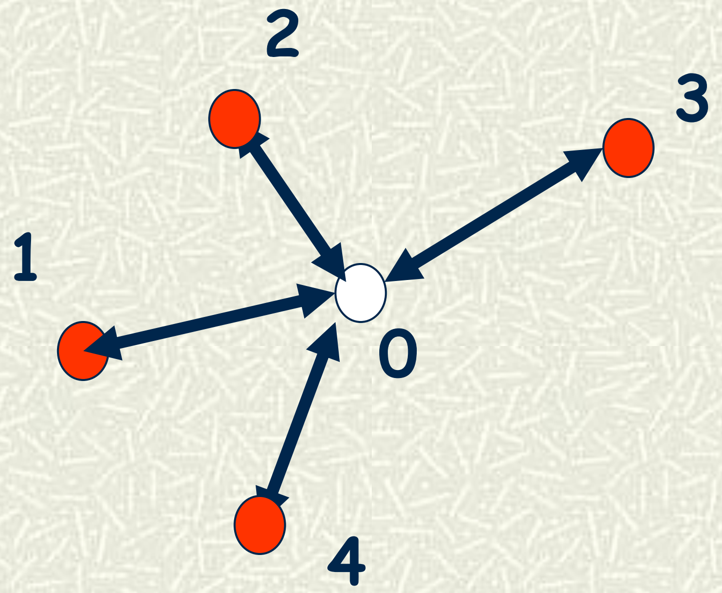
The n-hop Multilateration Problem

- Assumptions
  - Few of the nodes (if at all) are equipped with GPS (GPS-less)
  - A fraction of the nodes, called the beacons, are aware of their locations, others are referred as the unknowns
  - All the nodes within radio range of each other can measure the distance between each other

NORTHWESTERN UNIVERSITY

Recall…

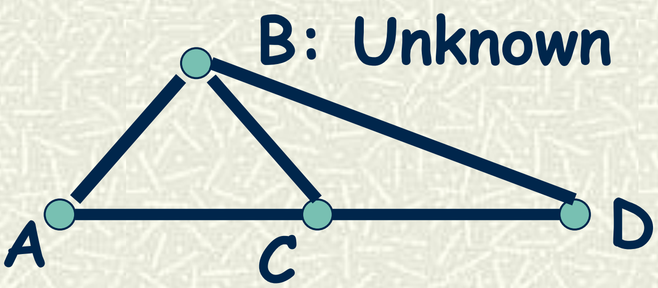
- Crucial 1-hop Multilateration Requirement
  - Within the range of at least three beacons



Next…

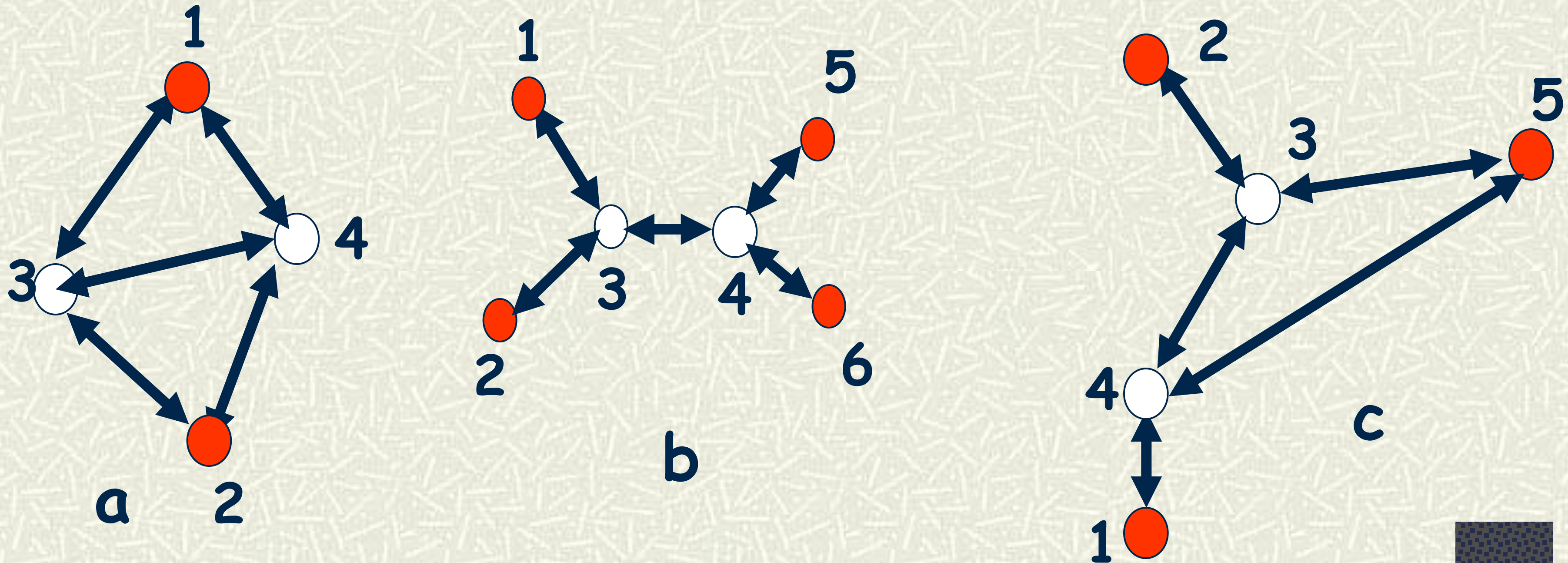
- Two Hop Multilateration Requirements
  - To have a unique possible position solution, it is necessary that an unknown node be connected to at least three nodes that have unique possible positions
  - It is necessary for an unknown node to use at least one reference point that is not collinear with the rest of its reference points

WHY?



Subsequently…

In a pair of unknown nodes using the link to each other as a constraint, each must have at least one link that connects to a different node from the nodes used as references by the other node



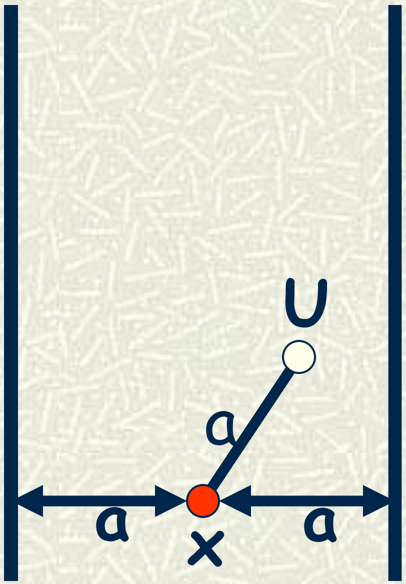
NORTHWESTERN UNIVERSITY

First Step in Multilatiration

- N-hop multilateration requirement
  - Have three neighbors that have unique positions?
  - Ask its unknown neighbor to determine its position
    - Assume the caller has tentatively unique solution
    - Meet the constraints
  - Do it recursively

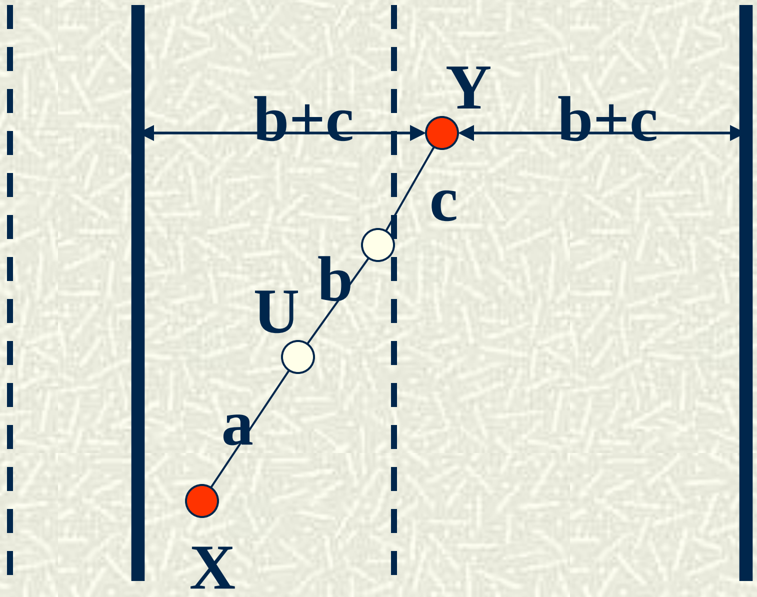
Step 2: Initial Estimates

- Use the accurate distance measurements to impose constraints in the x and y coordinates – **bounding box**
- Use the distance to a beacon as bounds on the x and y coordinates



Step 2: Multiple Initial Estimates

- Use the accurate distance measurements to impose constraints in the x and y coordinates – bounding box
- Do the same for beacons that are multiple hops away
  - Node Y in the Figure…
- Select the most constraining bounds



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

End of Step 2: Initial Estimates

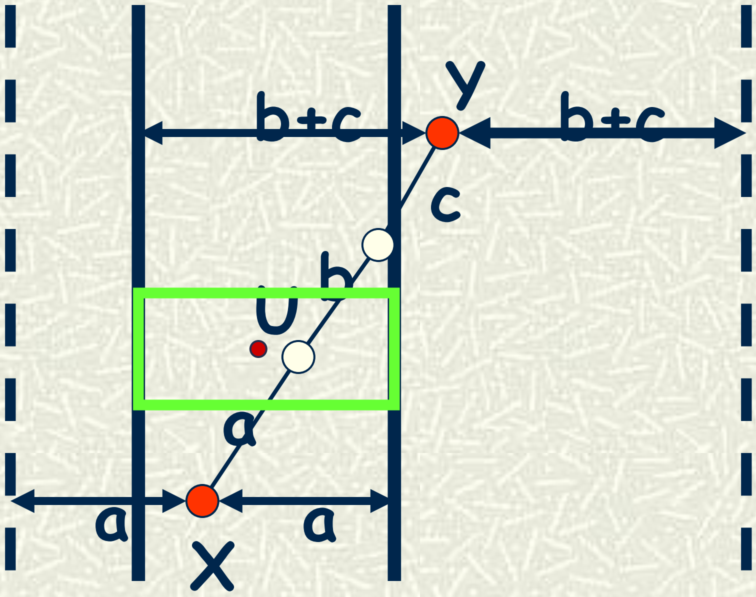
- Use the accurate distance measurements to impose constraints in the x and y coordinates
  - i.e., bounding box

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

- Do the same for beacons that are multiple hops away

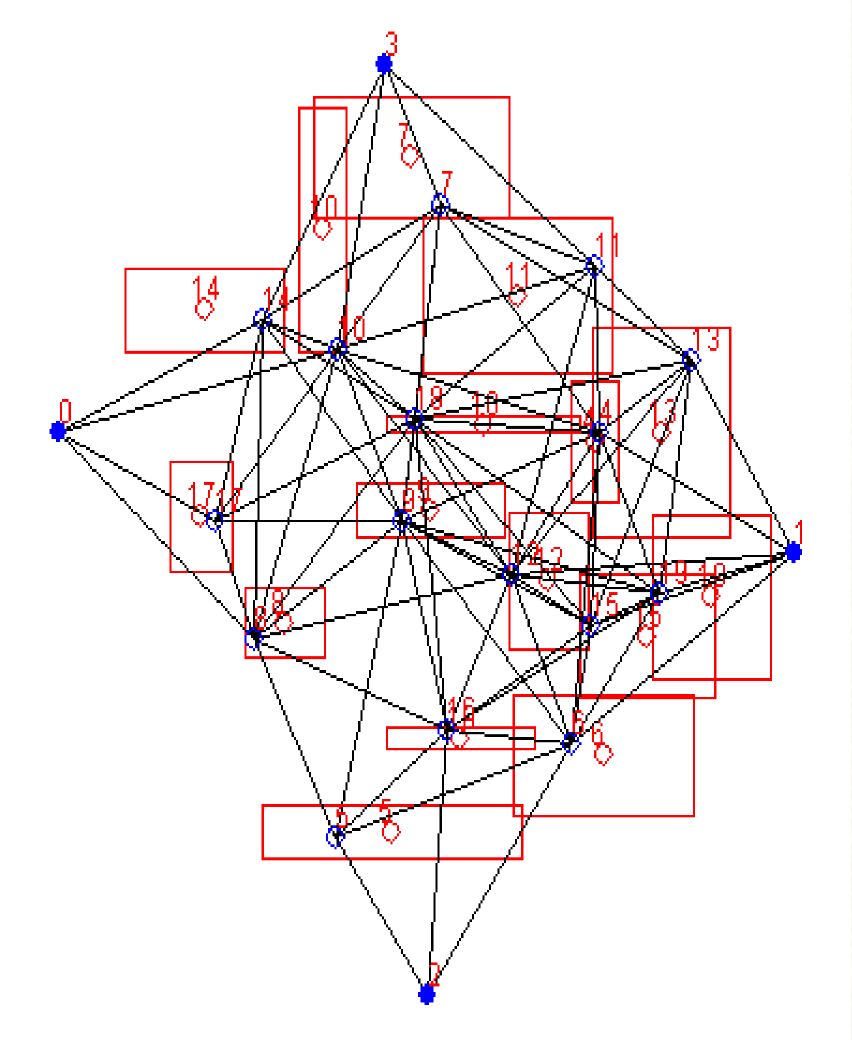
- Select the most constraining bounds

- Set the center of the bounding box as the initial estimate



End of Step 2: Globally

- Example:
  - 4 beacons
  - 16 unknowns
- To get good initial estimates, beacons should be placed on the perimeter of the network
- Observation: If the unknown nodes are outside the beacon perimeter then the initial estimates are on or very close to the convex hull of the beacons



Step 3: Refine the positions

- Use Kalman Filter to refine.
  - Start with prior info.
  - Incorporate new measurement info.
  - Improve the current state.

$$f_{i,j} = R_{i,j} - \sqrt{(ex_i - x_j)^2 + (ey_i - y_j)^2}$$

$$\sum f_{i,j}^2$$

$$\hat{x}^- = A\hat{x}_{k-1} + Bu_k \quad (1)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (2)$$

$$K = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (3)$$

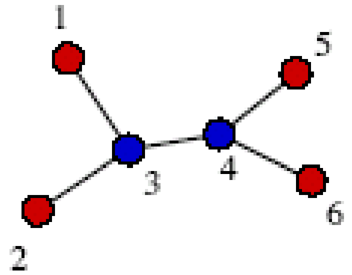
$$\hat{x}_k = K_k(z_k - H\hat{k})^{-1} \quad (4)$$

$$P_k = (I - K_k H)P_k^- \quad (5)$$

1. **Set the vector to the initial estimates**
2. **Evaluate equations 3,4 and 5 – the measurement update phase**
3. **Evaluate the convergence criterion**

Step 3: Example of Intuition…

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



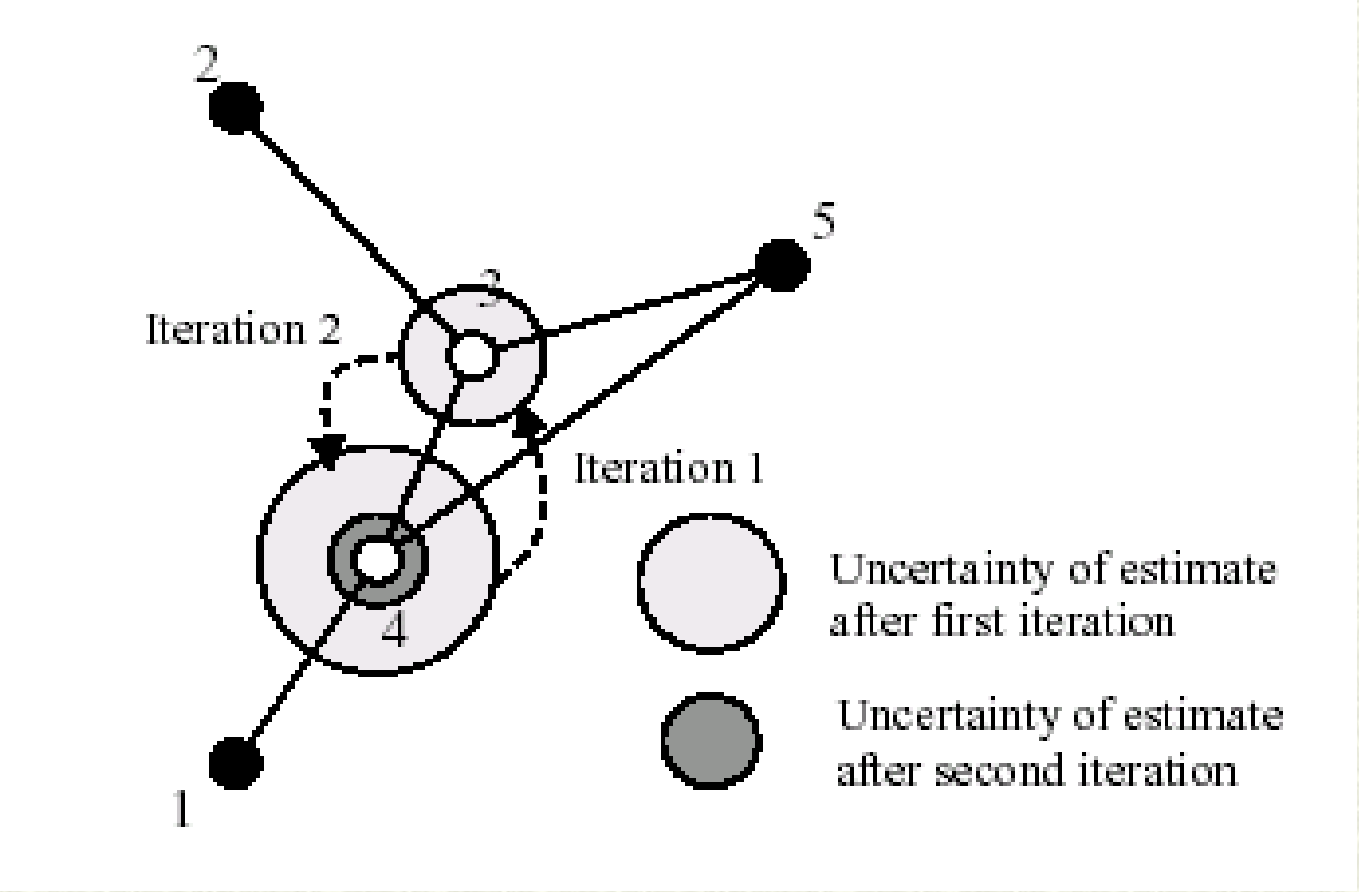
The objective function is

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

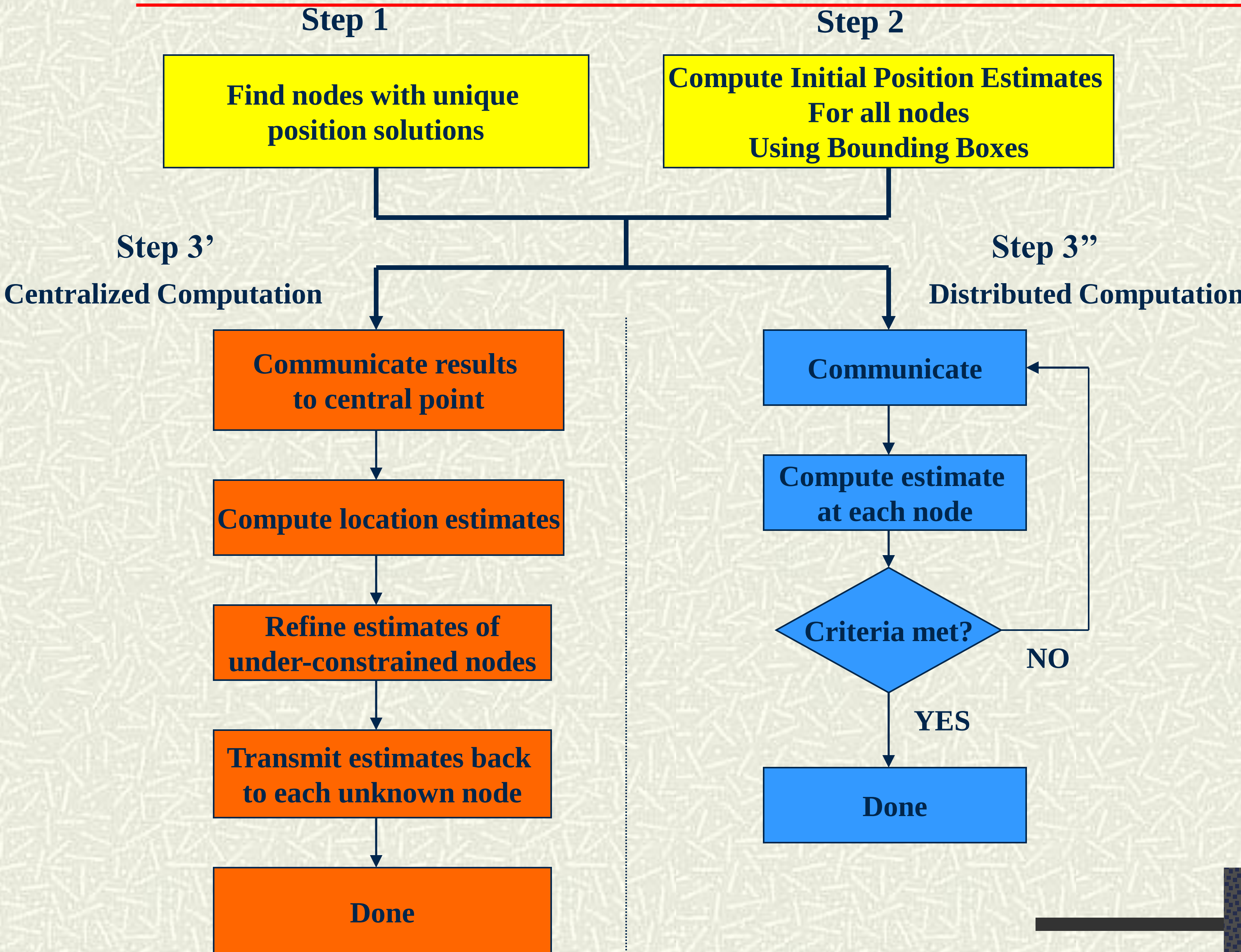
Start from some initial estimates, then use a Kalman Filter.

Step 3: Position Refinement (Distributed)

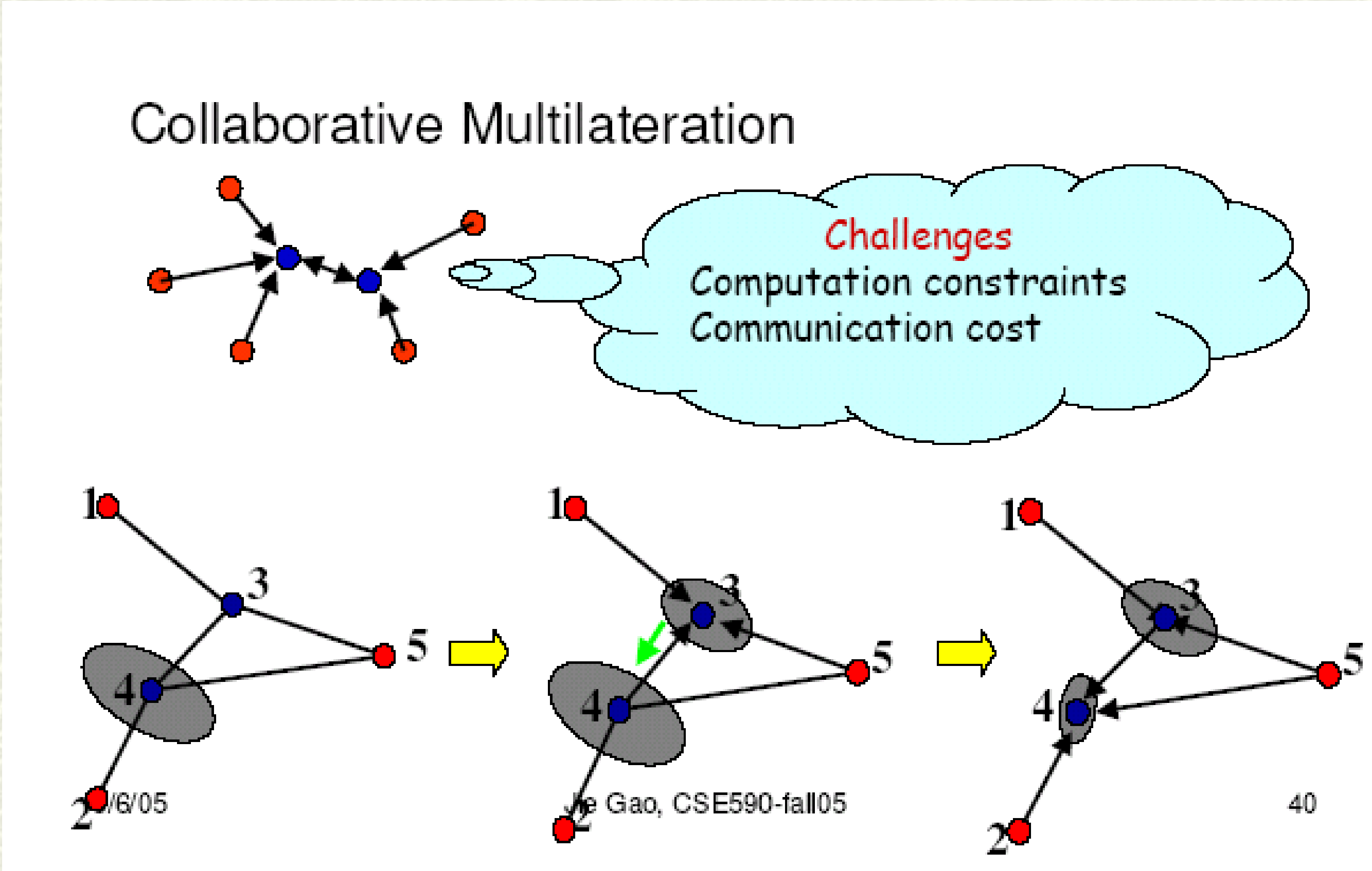
Kalman filters are computationally expensive;
Transmitting data to a centralized location ->
"communicationally" expensive!!!



In summary…


Step 1
Find nodes with unique position solutions
Step 2
Compute Initial Position Estimates For all nodes Using Bounding Boxes
Step 3'
Centralized Computation
Communicate results to central point
Compute location estimates
Refine estimates of under-constrained nodes
Transmit estimates back to each unknown node
Done
Step 3"
Distributed Computation
Communicate
Compute estimate at each node
Criteria met?
NO
YES
Done

In summary…



Channel effect + Detection Error + Setup Error…

In Summary…

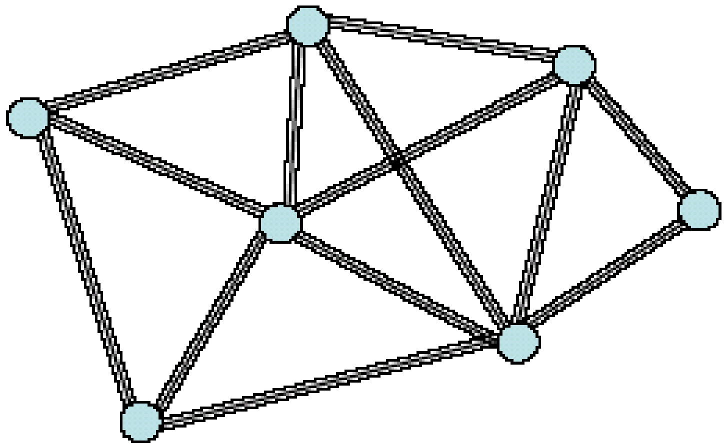
- Localization:
  - Still many open problems
  - Design decisions based on availability of technology, and constraints of the operating environment
    - Can we have powerful computation?
    - What is the availability of infrastructure support?
    - What type of obstructions are in the environment?
    - How fast, accurate, reliable should the localization process be?

(very domain-specific…)

Potpourri: Can the error- accumulation of the localization be improved???

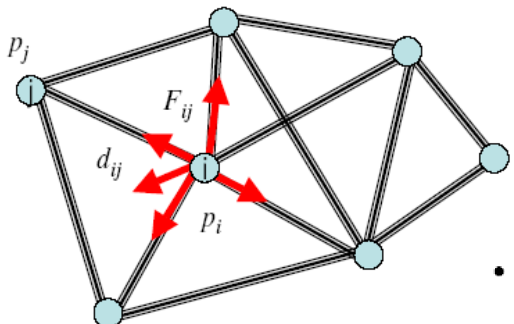
Mass-spring system

- Nodes are "masses", edges are "springs".
- Length of the spring equals the distance measurement.
- Springs put forces to the nodes.
- Nodes move.
- Until the system stabilizes.

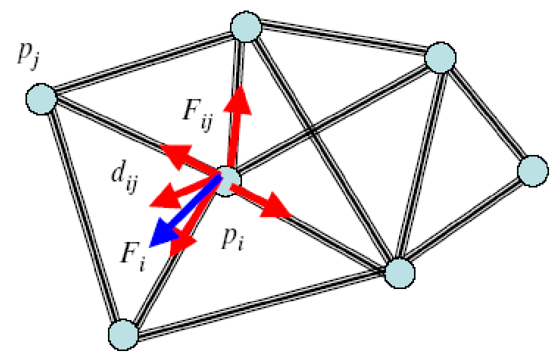


Potpourri: Mass-Spring Concepts

- Node n_i's current estimate of its position: p_i.
- The estimated distance d_{ij} between n_i and n_j.
- The measured distance r_{ij} between n_i and n_j.
- Force: $F_{ij} = d_{ij} - r_{ij}$, along the direction $p_i p_j$.

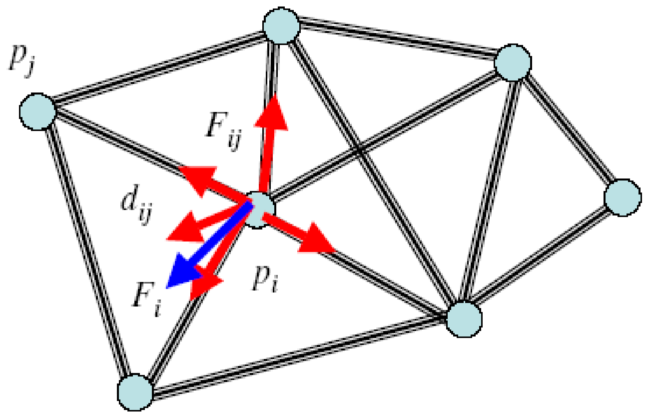


- Total force on n_i: $F_i = \Sigma F_{ij}$.
- Move the node n_i by a small distance (proportional to F_i).
- Recurse.



Potpourri: Mass-Spring Concepts

- Total energy n_i: $E_i = \Sigma E_{ij} = \Sigma (d_{ij} - r_{ij})^2$.
- Make sure that the total energy $E = \Sigma E_i$ goes down.
- Stop when the force (or total energy) is small enough.

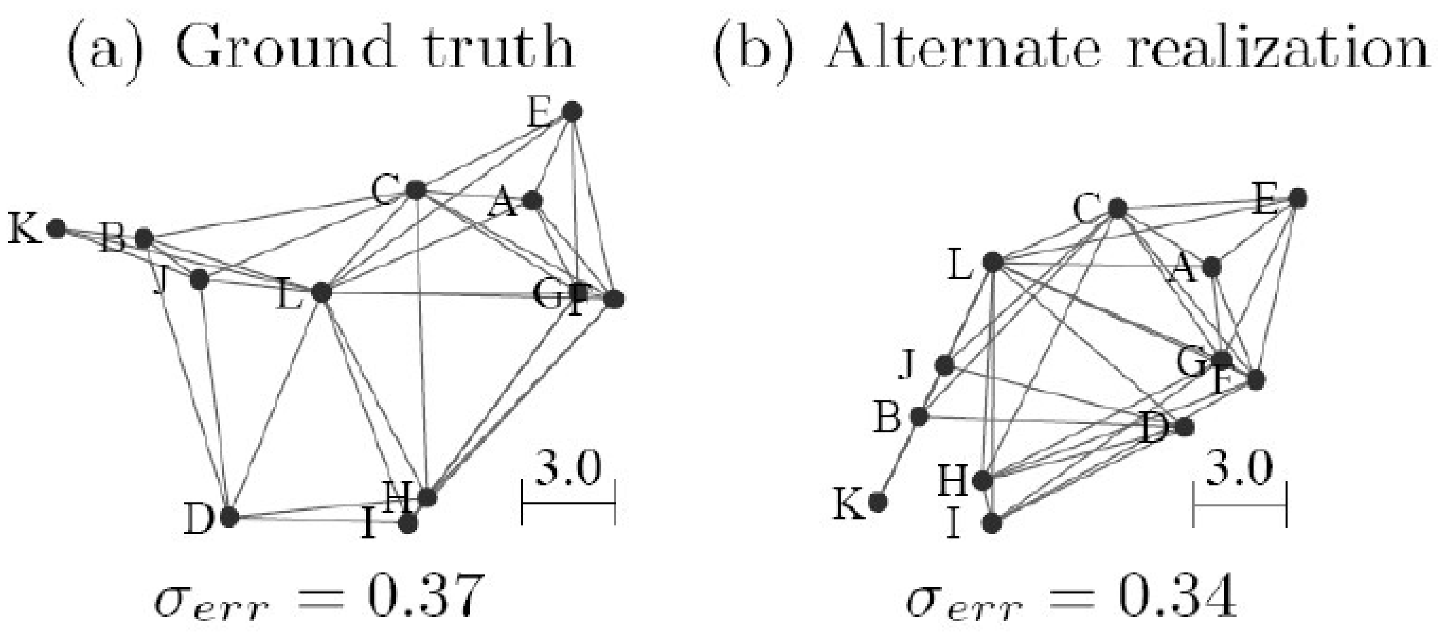


- A distributed algorithm.
- Problem: may stuck in local minima.
- Still need to start from a reasonably good initial estimation, e.g., the iterative multi-lateration.

Potpourri: Mass-Spring *Problem*…

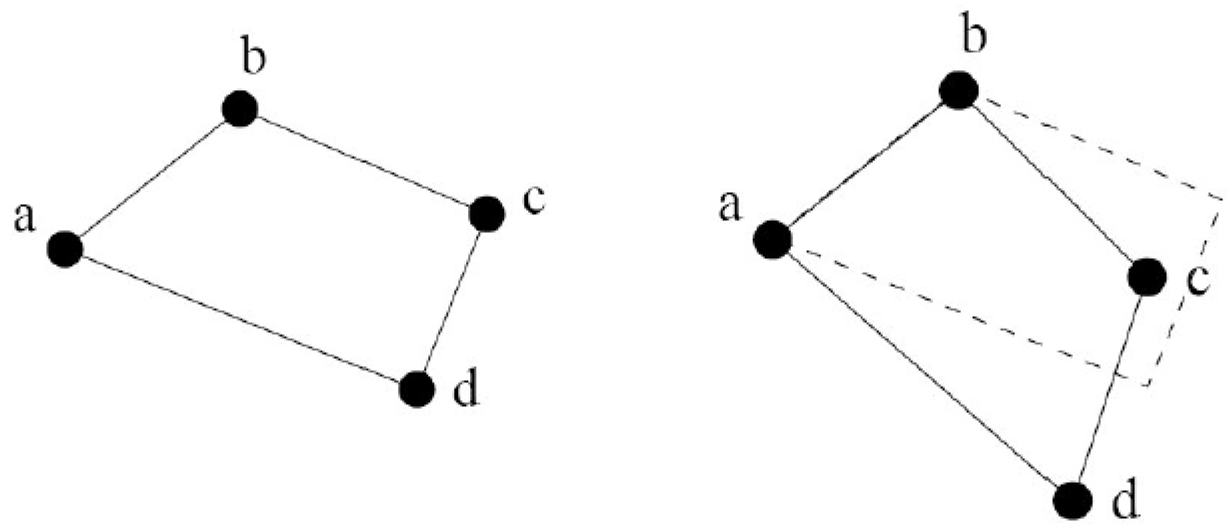
Optimization doesNOT solve the ***ambiguity*** of the localization (e.g. noise in the measurements…)

- Same distances, different realization.



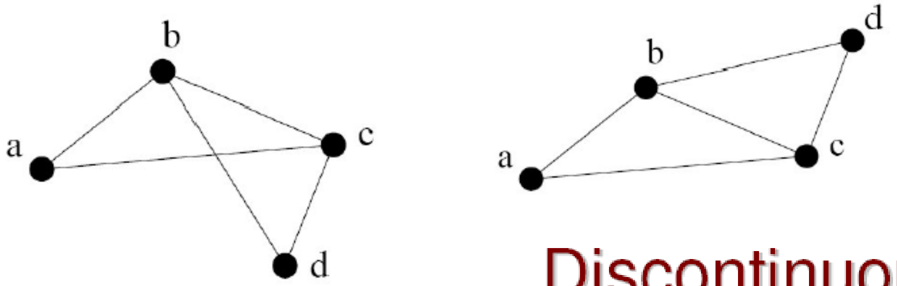
Porpourri: Mass-Spring Problems – Examples

- Nodes move continuously without violating the distance constraints.



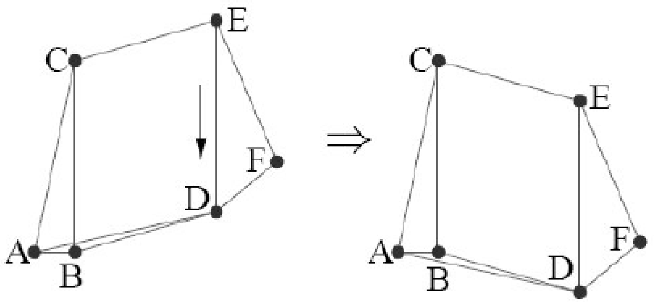
Potpourri: Mass-Spring Problems – More Examples…

- No continuous deformation, but subjects to global flipping.



Discontinuous flex ambiguity

- Remove AD, flip ABD up, insert AD.
- No continuous deformation in between.
- But both are valid realization of the distances.



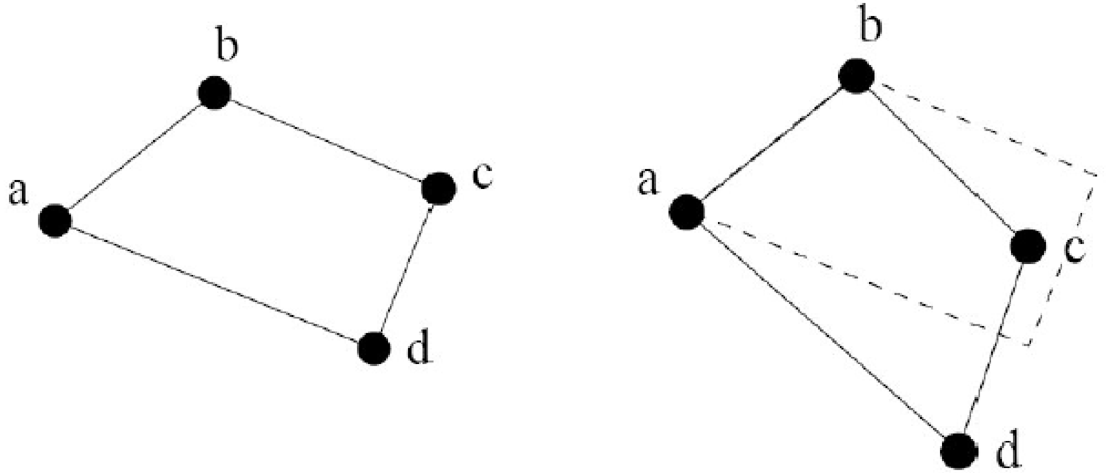
Potpourri: Solution to the Mass-Spring-like Problems

General Questions: Given a set of distance-measurements, does there exist a unique graph representing the actual reality? If not, what is the best that we can do? What are the criteria? How easy/hard is it to check/verify them?

Rigidity theory

Tool:

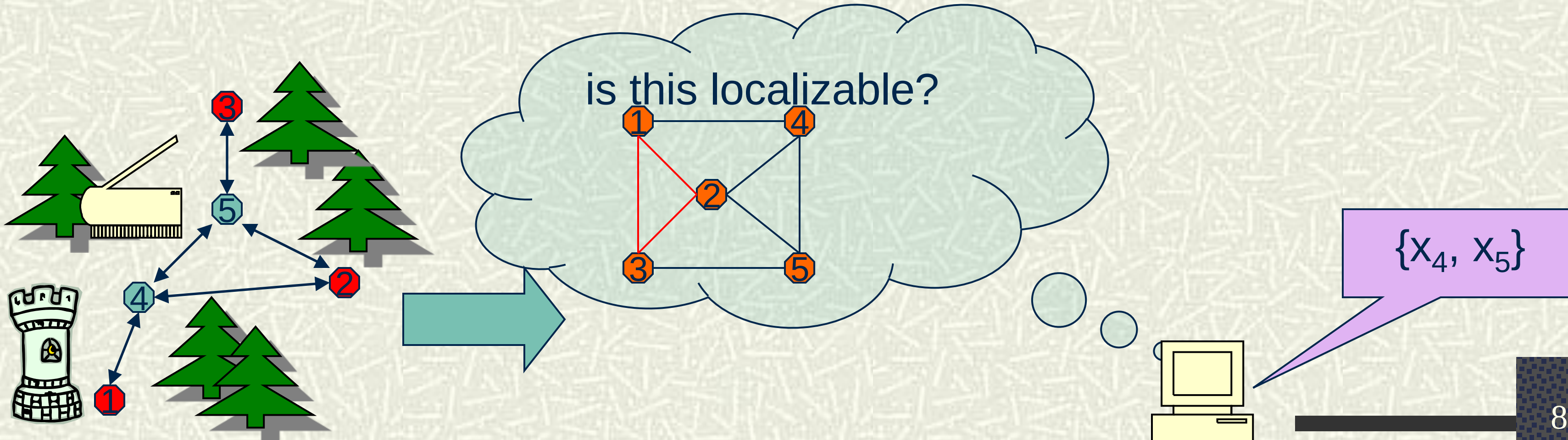
- Given a set of rigid bars connected by hinges, rigidity theory studies whether you can move them continuously.



Potpourri: Example…

- Problem is solvable if there is exactly one set of points $\{x_{m+1},\ldots,x_n\}$ (one *realization* of the graph) consistent with G_n, $\{x_1,\ldots,x_m\}$ and

  $d{:}E \rightarrow \mathbf{R}$
- The solvability of the problem does *not* depend on the positions of the points
- Graph properties alone (!) of a *variant* of G_n that is called the *grounded graph* determine if the problem is solvable with probability 1
  - By adding all edges between beacons to G_n, we get the *Grounded Graph*

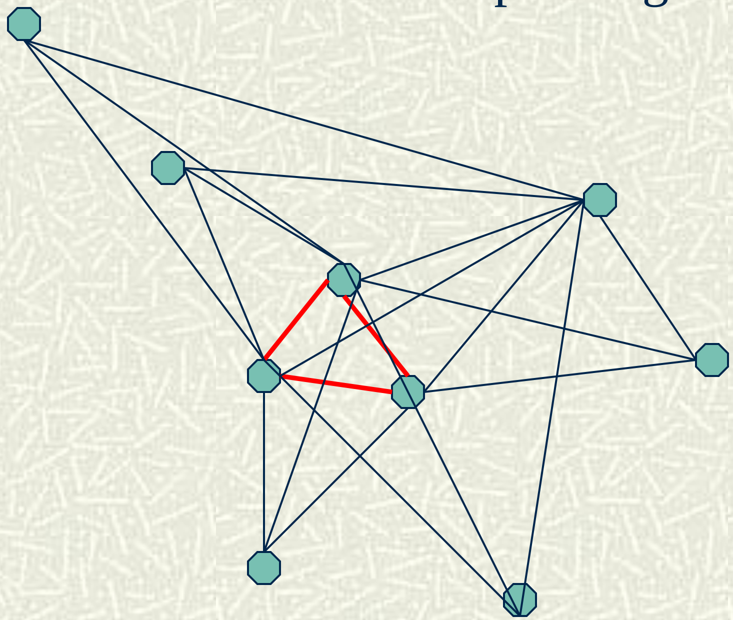


Potpourri: Hardness Results

Network localization is solvable iff its associated *grounded graph* is redundantly rigid and 3-connected (graph of network connectivity, augmented with edges between all pairs of beacons).

Theorem: Realization of uniquely realizable graphs weighted so as to be realizable is NP-hard.

- There are subclasses of globally rigid graphs that are easy to localize.
- For instance *trilateration graphs* – those graphs obtained through a sequence of extensions corresponding to trilaterations from a triangle:



Trilateration graphs are:
- Globally rigid
- Realizable in polynomial time

Readings…

- A. Savvides, L. Girod, M. Srivastava, and D. Estrin, "**Localization in Sensor Networks**," Book Chapter in: *Wireless Sensor Networks*, Edited by Znati, Radhavendra and Sivalingam, Kluwer Academic Publishers, 2004.
- B. Sundararaman, U. Buy, A. D. Kshemkalyani, "**Clock synchronization for wireless sensor networks: a survey**" Ad Hoc Networks 3 (2005) 281–323, 2005
  - **Additional/Recommended:**
  - Lecture notes on Rigidity Theory by Prof. Jie Gao (Dept. of Computer Science, SUNY at Stony Brook)
  - "Mobile-Assisted Localization in Wireless Sensor Networks", N.B. Priyantha, H. Balakrishnan, E. Demaine, S. Teller, IEEE INFOCOM 2005.