# Pervasive Systems

Ioannis Chatzigiannakis

Sapienza University of Rome
Department of Computer, Control, and Management Engineering (DIAG)

Lecture 7:
Localization & Locating Systems





## Position of Users

- Many applications require the location of the user to provide full set of services
  - Fundamental problem in Pervasive systems
- Depending on the operating conditions the problem is hard to solve:
  - indoor/outdoor,
  - low energy,
  - low communication overhead.
- Many algorithms proposed by researchers
- Easier to solve under certain conditions





## Localization vs Location Services

- Location Systems identify and track the location of objects or people.
  - Usually in real-time.
- Location Directory Services:
  - A Particular Node,
  - Person,
  - Data,
  - Equipment,
  - Resource,
  - Service.





## Node Localization

- Means for a node to determine its position,
  - with respect to some coordinate system.
  - coordinate system may be related to the physical world,
  - or it may be a symbolic location system.
- Using the help of
  - Anchor nodes that know their position,
  - Directly adjacent,
  - Over multiple hops.

# Node Localization

- Centralized or distributed computation
- Scale (indoors, outdoors, global, . . . )
- Metrics
  - Accuracy (how close is an estimated position to the real position?)
  - Precision (for repeated position determinations, how often is a given accuracy achieved?)
  - Costs, energy consumption, . . .





# Taxonomy of Localization Techniques

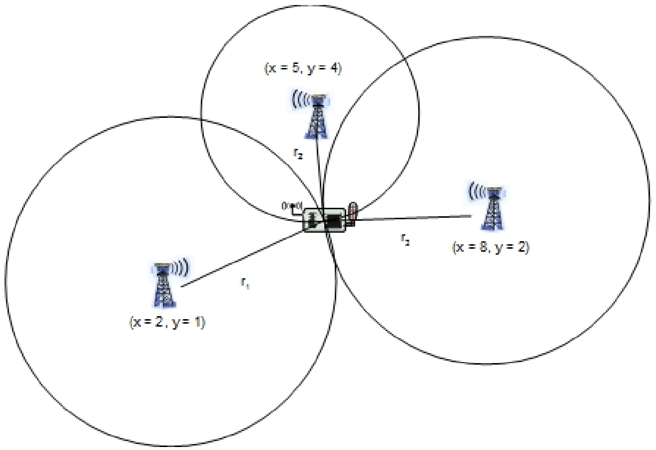
1. Range Based
   - Determine distances between nodes (range)
   - Then compute location using geometry
2. Range Free
   - No need to determine distances directly, use other techniques.
   - e.g., use hop count:
     - Use average distances between hops
     - Then compute location using geometry





# Proximity

- Exploit finite range of wireless communication
- E.g.: easy to determine location in a room with infrared room number announcements

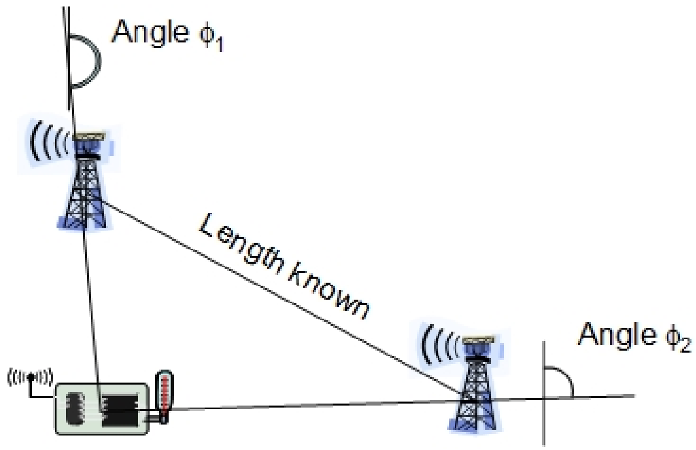







# (Tri-/Multi-)lateration and angulation

- Use distance or angle estimates, simple geometry to compute position estimates



# Scene-based

- Perform a "scene analysis"
- Radio environment has characteristic "signatures"
- Measured beforehand and store,
- compared with current situation.





# Estimating distances using RSSI

- RSSI: Received Signal Strength Indicator
- Translate signal strength into distance
  - Use model/formula to do the conversion
  - E.g., signal strength drops as inverse square of distance
- Send out signal of known strength, use received signal strength and path loss coefficient to estimate distance

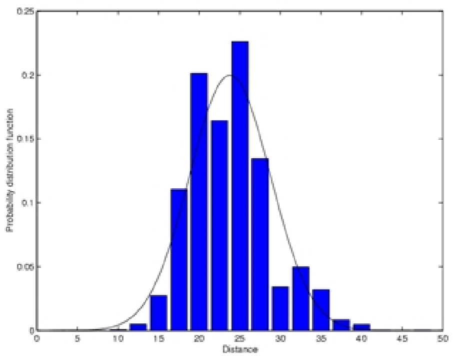
$$P_{recv} = c\frac{P_{tx}}{d^{\alpha}} \Leftrightarrow d = \sqrt[\alpha]{\frac{cP_{tx}}{P_{recv}}}$$

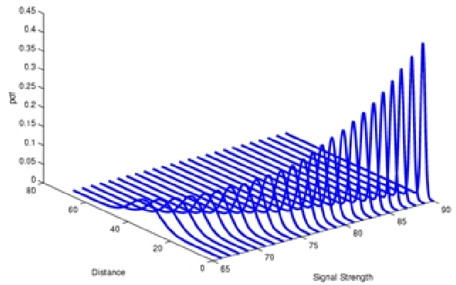




# Estimating distances using RSSI

- Highly error-prone process
- Technique largely unsuitable:
  Multi-path fading, background interference, irregular signal propagation ...



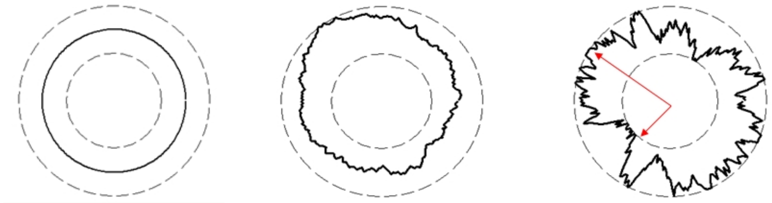





# Continuous Radio Variation Model

**Degree of Irregularity (DOI)**

Defined as maximum radio range variation per unit degree change in the direction of radio propagation.



DOI=0 DOI=0.05 DOI=0.2

# Link Quality Indicator (LQI)

- Another metric of the current quality of the received signal.

LQI measures the received energy and/or SNR for each received packet. When energy level and SNR information are combined, they can indicate whether a corrupt packet resulted from low signal strength or from high signal strength plus interference.

How easily a received signal can be demodulated by accumulating the magnitude of the error between ideal constellations and the received signal over the 64 symbols immediately following the sync word?





# LQI vs RSSI

- RSSI is a signal strength indication
  - It does not care about the "quality" or "correctness" of the signal.
- LQI does not care about the actual signal strength,
  - it is a relative measurement of the link quality: a low value indicates a better link than what a high value does.
  - but the signal quality often is linked to signal strength.
  - a strong signal is likely to be less affected by noise and thus will be seen as "cleaner" or more "correct" by the receiver.





# LQI vs RSSI

"Extreme cases" to illustrate how RSSI and LQI work:

1. A weak signal in the presence of noise may give low RSSI and high LQI.
2. A weak signal in "total" absence of noise may give low RSSI and low LQI.
3. Strong noise (usually coming from an interferer) may give high RSSI and high LQI.
4. A strong signal without much noise may give high RSSI and low LQI.
5. A very strong signal that causes the receiver to saturate may give high RSSI and high LQI.





# Packet Reception Rate (PRR)

- PRR is approximated as the probability of successfully receiving a packet between two neighbor nodes.
- If PRR is high that means the link quality is high and vice versa.
- A receiver centric reliability index evaluating how a broadcast packet from a sender is received by all intended receivers.
- For a given sender node:

$$PRR = \frac{\text{No. of nodes receiving a packet}}{\text{Total no. of nodes in the transmission range}}$$

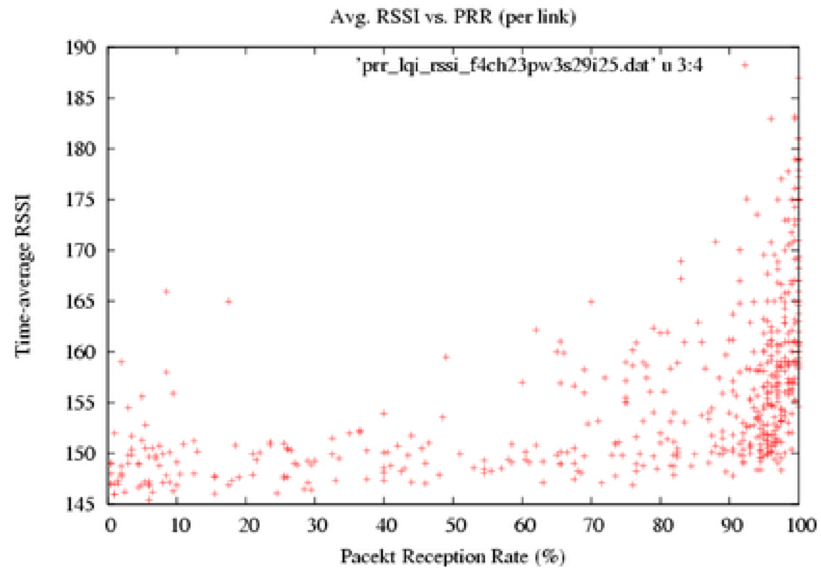
# Relationship between PRR, LQI, and RSSI

- IEEE 802.15.4
- Payload-length 29bytes
- Tx-interval 25ms
- Channel 23, Power 3





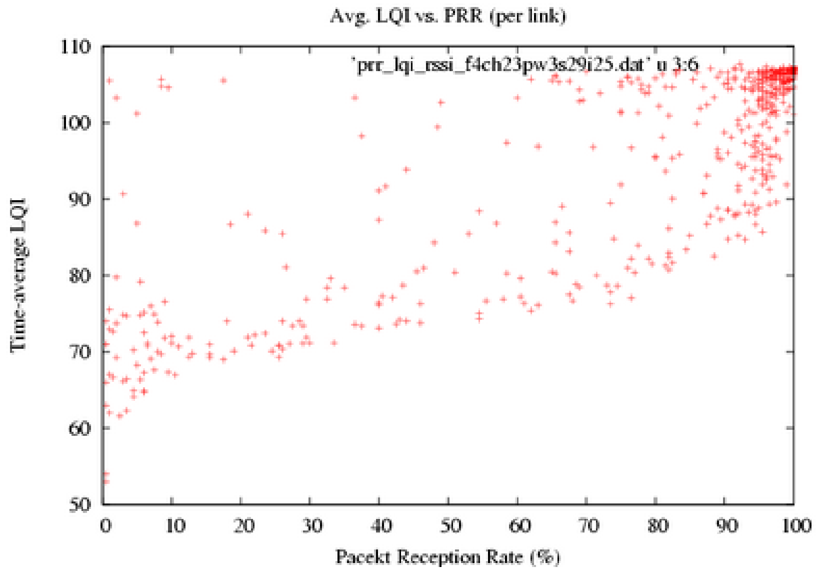
Relationship between PRR, LQI, and RSSI







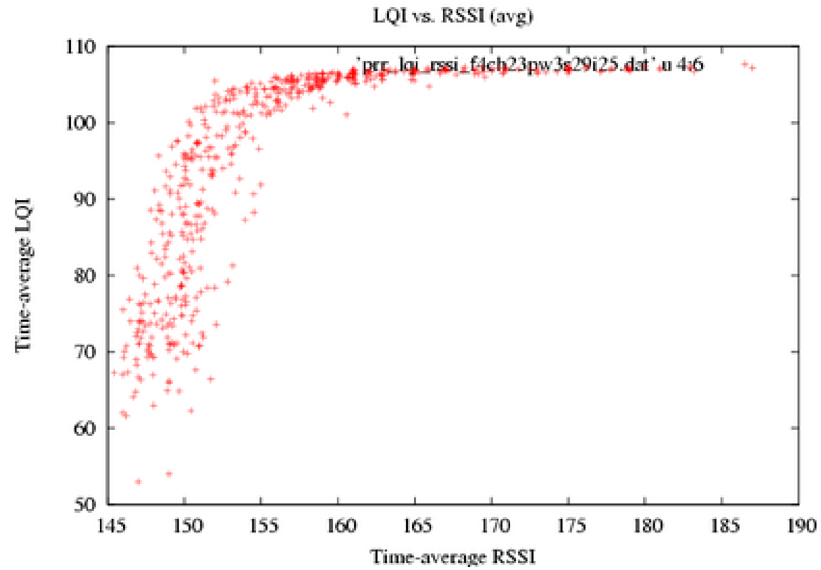

Relationship between PRR, LQI, and RSSI








Relationship between PRR, LQI, and RSSI



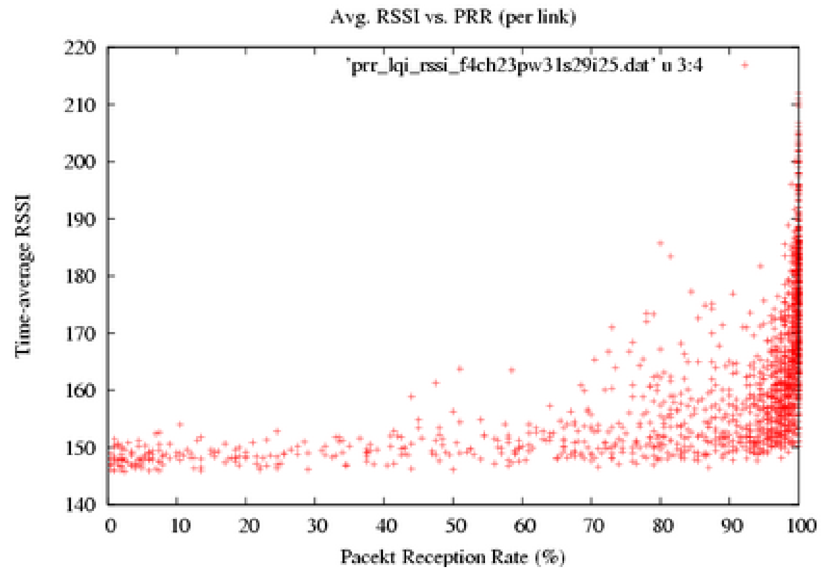
# Relationship between PRR, LQI, and RSSI

- IEEE 802.15.4
- Payload-length 29bytes
- Tx-interval 25ms
- Channel 23, Power 31





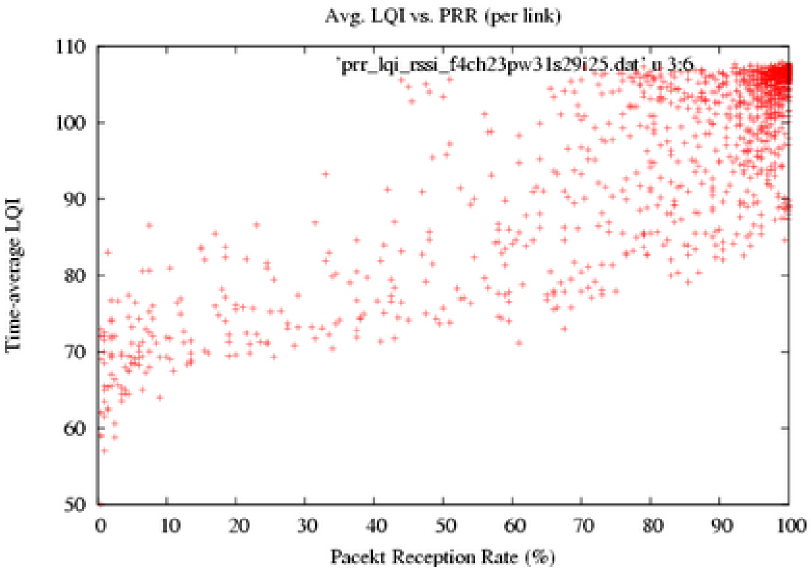
# Relationship between PRR, LQI, and RSSI







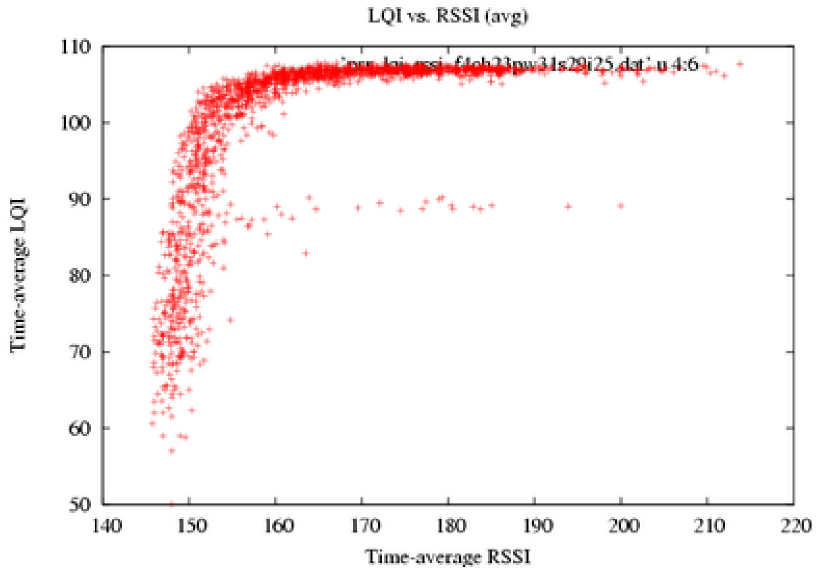

# Relationship between PRR, LQI, and RSSI








# Relationship between PRR, LQI, and RSSI



# Time of arrival (ToA)

- Use time of transmission, propagation speed, time of arrival to compute distance
- Distance between sender and receiver of a signal can be determined using the measured signal propagation time and known signal velocity.
- sound waves: $343 m/s$, i.e., approx. $30 ms$ to travel $10 m$
- radio signals: $300 km/s$, i.e., approx. $30 ns$ to travel $10 m$
- Problem: Exact time synchronization





# ToA – GPS

- Constellation of 27 satellites – 24 active and 3 redundant
- Clocks must be synchronized (use signal and clock to compute distance)
- Requires line of sight
- Issues:
  - Billions of dollars of infrastructure
  - Each node with GPS require increased battery power
  - May also be a problem with form factor – increases hardware size





# ToA – GPS

21 SATELLITES WITH 3 OPERATIONAL SPARES,
6 ORBITAL PLANES, 55 DEGREE INCLINATIONS
20,200 KILOMETER, 12 HOUR ORBITS

- Satellites are uniformly distributed in six orbits (4 satellites per orbit)
- Satellites circle earth twice a day at approx. 7000 miles/hour
- At least 8 satellites can be seen simultaneously from almost anywhere





# ToA – GPS

- Each satellite broadcasts coded radio waves (pseudorandom code), containing
  1. identity of satellite
  2. location of satellite
  3. the satellites status
  4. data and time when signal was sent
- Six monitor stations constantly receive satellite data and forward data to a master control station (MCS)
- MCS is located near Colorado Springs, Colorado
- MCS uses the data from monitor stations to compute corrections to the satellites orbital and clock information which are sent back to the satellites

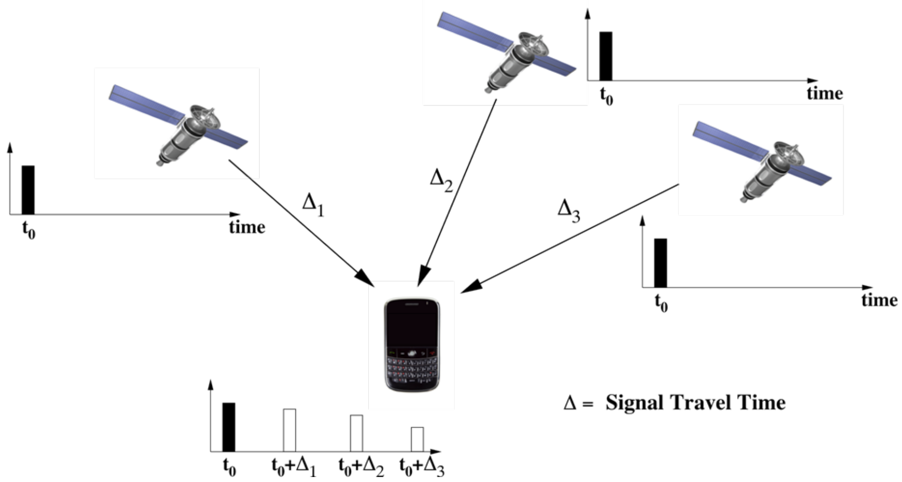
# ToA – GPS

- Satellites and receivers use accurate and synchronized clocks
- Receiver compares generated code with received code to determine
  1. the actual code generation time of the satellite
  2. time difference Δ between code generation time and current time
  3. Δ expresses the travel time of the code from satellite to receiver





# ToA – GPS








# ToA – GPS

- Radio waves travel at the speed of light (approx. 186,000 miles/second)
- With known Δ, the distance can be determined
- Receiver knows that it is located somewhere on a sphere centered on the satellite with a radius equal to this distance
- With three satellites, the location can be narrowed down to two points
  - typically one of these two points can be eliminated easily





# ToA – GPS

- With four satellites, accurate localization is possible
  - accurate positioning relies on accurate timing
  - receiver clocks are much less accurate than atomic GPS clocks
  - small timing errors lead to large position errors
    - example: clock error of 1ms translates to a position error of 300km
  - fourth sphere would ideally intersect with all three other spheres in one exact location
  - spheres too large: reduce them by adjusting the clock (moving it forward)
  - spheres too small: increase them by adjusting the clock (moving it backward)

# ToA – GPS

- Most GPS receivers today can achieve good accuracy (e.g., 10m or less)
- Additional advanced techniques can be used to further improve accuracy:
- example: Differential GPS (DGPS)
  - relies on land-based receivers with exactly known locations
  - they receive signals, compute correction factors, and broadcast them to GPS receivers
  - GPS receivers correct their own measurements





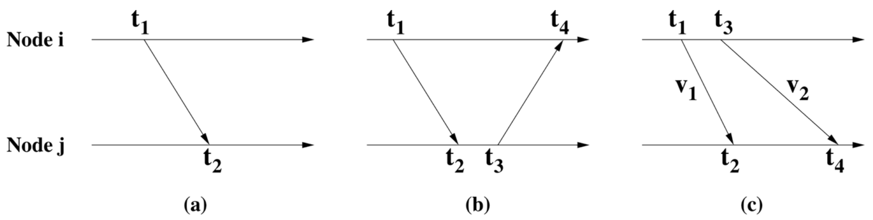
# Time Difference of Arrival (TDoA)

- Use two different signals with different propagation speeds
- Example: ultrasound and radio signal
  Propagation time of radio negligible compared to ultrasound
- Speed of sound is much slower (approximately 331.4$m/s$) than radio
- Compute difference between arrival times to compute distance
- Problem: Calibration, expensive/energy-intensive hardware





# Time Difference of Arrival (TDoA)




- example: radio signal (sent at $t_1$ and received at $t_2$), followed by acoustic signal (sent at $t_3 = t_1 + t_{wait}$ and received at $t_4$)
- no clock synchronization required
- distance measurements can be very accurate
- need for additional hardware

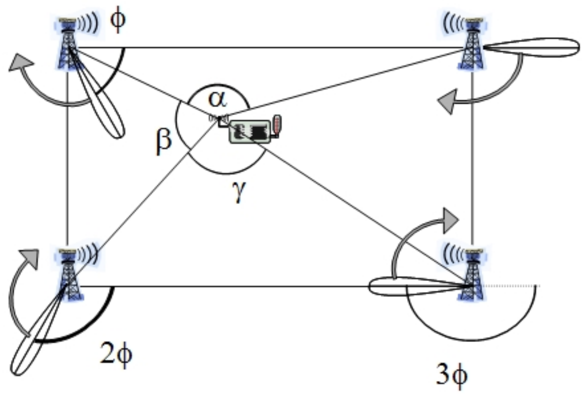
$$dist = (v_1 - v_2) \times (t_4 - t_2 - t_{wait})$$





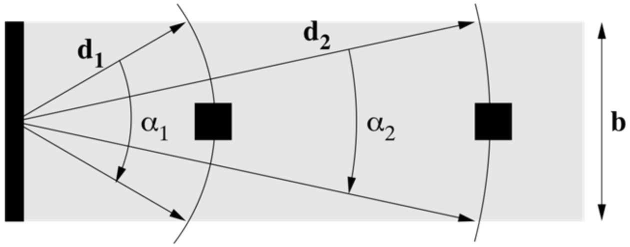
# Determining Angles – Directional antennas

- On the node: Mechanically rotating or electrically “steerable”
- On several access points
  Rotating at different offsets Time between beacons allows to compute angles



# Lighthouse: an event-driven localization approach

- Requirement: base station with light emitter
- Idealistic light source: emitted beam of light is parallel (constant width $b$)
- Light source rotates such that node sees beam of light for $t_{beam}$




$$d = \frac{b}{2\sin\left(\frac{\alpha}{2}\right)} \qquad \alpha = 2\pi\frac{t_{beam}}{t_{turn}}$$





# Lighthouse: an event-driven localization approach

- Perfectly parallel light beams are hard to realize in practice
- Small beam spreads can result in large localization errors
  - if $b = 10cm$ and $spread = 1°$, $b = 18.7cm$ at $5m$ distance
- Beam width should be large to keep inaccuracies small
- Solution: two laser beams that outline a "virtual" parallel beam
  - only edges of the virtual beam are of interest





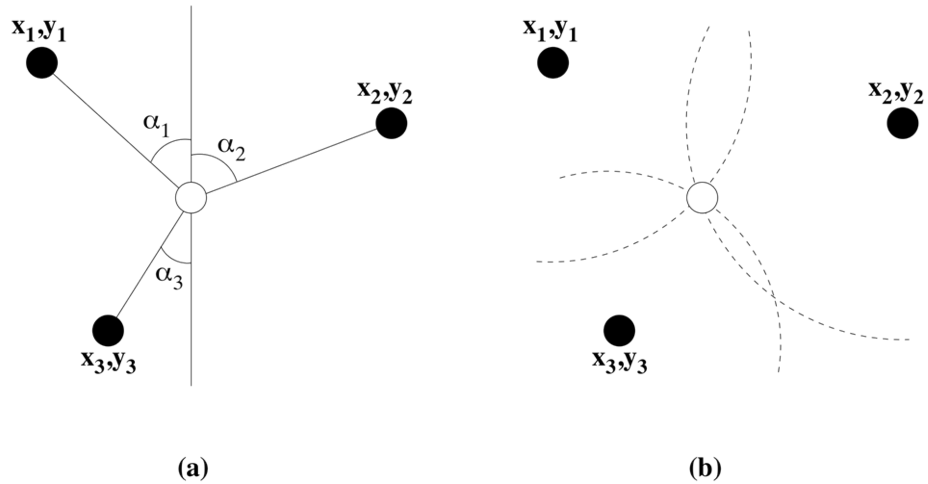
# Trilateration

- Localization based on measured distances between a node and a number of anchor points with known locations
- Basic concept: given the distance to an anchor, it is known that the node must be along the circumference of a circle centered at anchor and a radius equal to the node-anchor distance
- In two-dimensional space, at least three non-collinear anchors are needed and in three-dimensional space, at least four non-coplanar anchors are needed





# Trilateration



# Trilateration

- Assuming distances to three points with known location are exactly given
- Solve system of equations (Pythagoras!)
  - $(x_i, y_i)$ : coordinates of anchor point $i$, $r_i$ distance to anchor $i$
  - $(x_u, y_u)$ : unknown coordinates of node

$$(x_i - x_u)^2 + (y_i - y_u)^2 = r_i^2 \quad \textit{for } i = 1, \ldots, 3$$

  - Subtracting eq. 3 from 1 & 2:

$$(x_1 - x_u)^2 - (x_3 - x_u)^2 + (y_1 - y_u)^2 - (y_3 - y_u)^2 = r_1^2 - r_3^2$$

$$(x_2 - x_u)^2 - (x_3 - x_u)^2 + (y_2 - y_u)^2 - (y_3 - y_u)^2 = r_2^2 - r_3^2$$





# Trilateration

- Rearranging terms gives a linear equation in $(x_u, y_u)$!

$$2(x_3 - x_1)x_u - 2(y_3 - y_1)y_u = (r_1^2 - r_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2)$$

$$2(x_3 - x_2)x_u - 2(y_3 - y_2)y_u = (r_2^2 - r_3^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2)$$





# Trilateration as matrix equation

- Rewriting as a matrix equation:

$$2\begin{bmatrix} x_3 - x_1 & y_3 - y_1 \\ x_3 - x_2 & y_3 - y_2 \end{bmatrix}\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} (r_1^2 - r_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) \\ (r_2^2 - r_3^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2) \end{bmatrix}$$





# Example

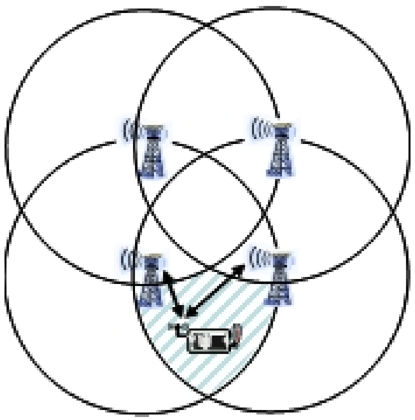
- $(x_1, y_1) = (2, 1)$
- $(x_2, y_2) = (5, 4)$
- $(x_3, y_3) = (8, 2)$
- $r_1 = 10^{0.5},\ r_2 = 2,\ r_3 = 3$

$$2\begin{bmatrix} 6 & 1 \\ 3 & -2 \end{bmatrix}\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} 64 \\ 22 \end{bmatrix}$$

- $!(x_u, y_u) = (5, 2)$

# Overlapping connectivity

- Position is estimated.
- In the center of area where circles from which signal is heard/not heard overlap.

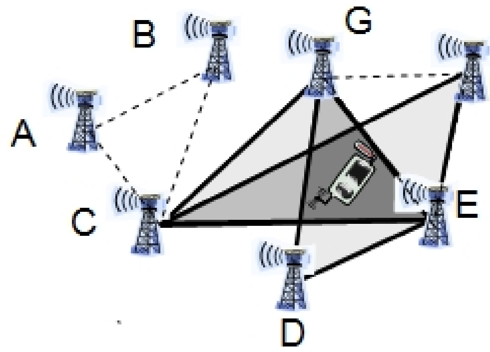






# Approximate point in triangle

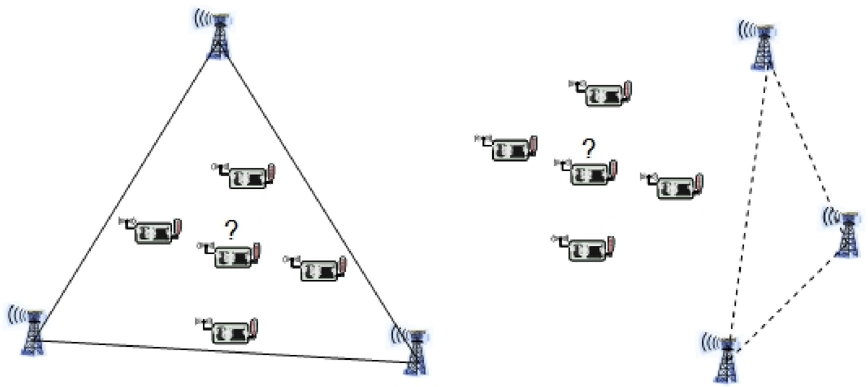
- Determine triangles of anchor nodes where node is inside, overlap them.
- Check whether inside a given triangle – move node or simulate movement by asking neighbors.
- Only approximately correct








# Approximate point in triangle

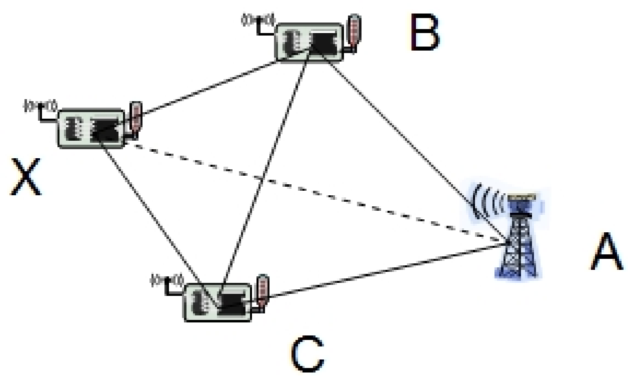







# Multihop range estimation

- How to estimate range to a node to which no direct radio communication exists?
  - No RSSI, TDoA, ...
  - But: Multihop communication is possible



# Two Ideas

- Step 1: DV-Hop
  - Count number of hops, assume length of one hop is known.
  - Start by counting hops between anchors, divide known distance.
- Step 2: DV-Distance
  - If range estimates between neighbors exist, use them to improve total length of route estimation in previous method





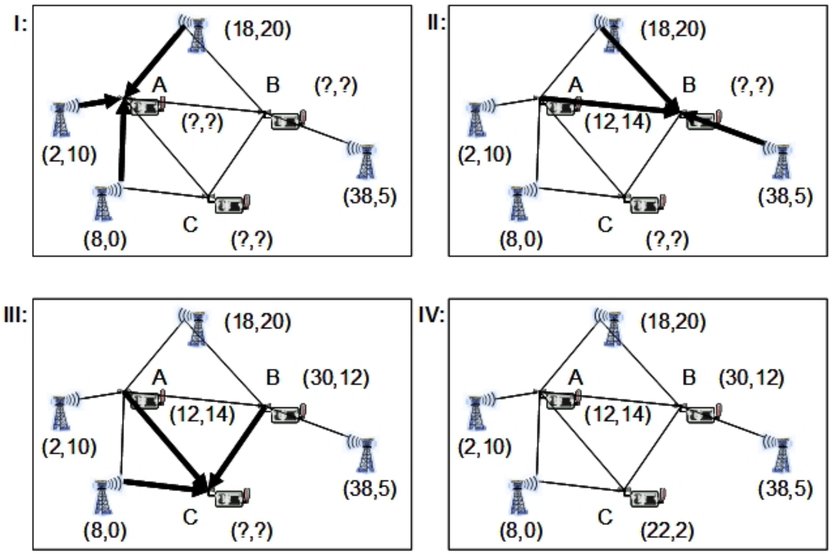
# Iterative multilateration

- Assume some nodes can hear at least three anchors (to perform triangulation), but not all
- Idea: let more and more nodes compute position estimates, spread position knowledge in the network
- Problem: Errors accumulate





# Iterative multilateration








# SpotLight

- Nodes randomly deployed from UAV/helicopter
- Nodes self-organize into a network, execute a time-sync protocol
- The UAV (Spotlight device) flies over the network and generates (invisible) light events
- Nodes detect the events and report the timestamps
- The Spotlight device computes the location of the nodes
- No extra hardware needed on nodes!

# SpotLight