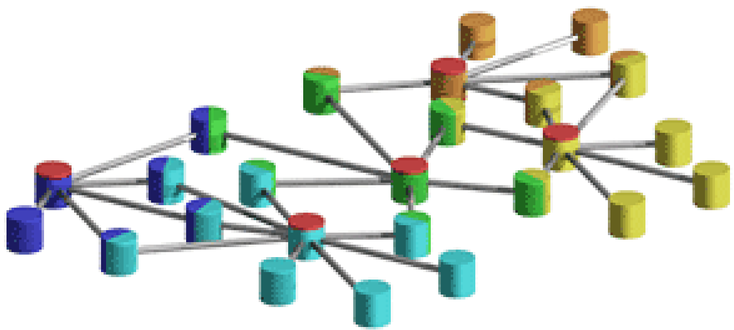
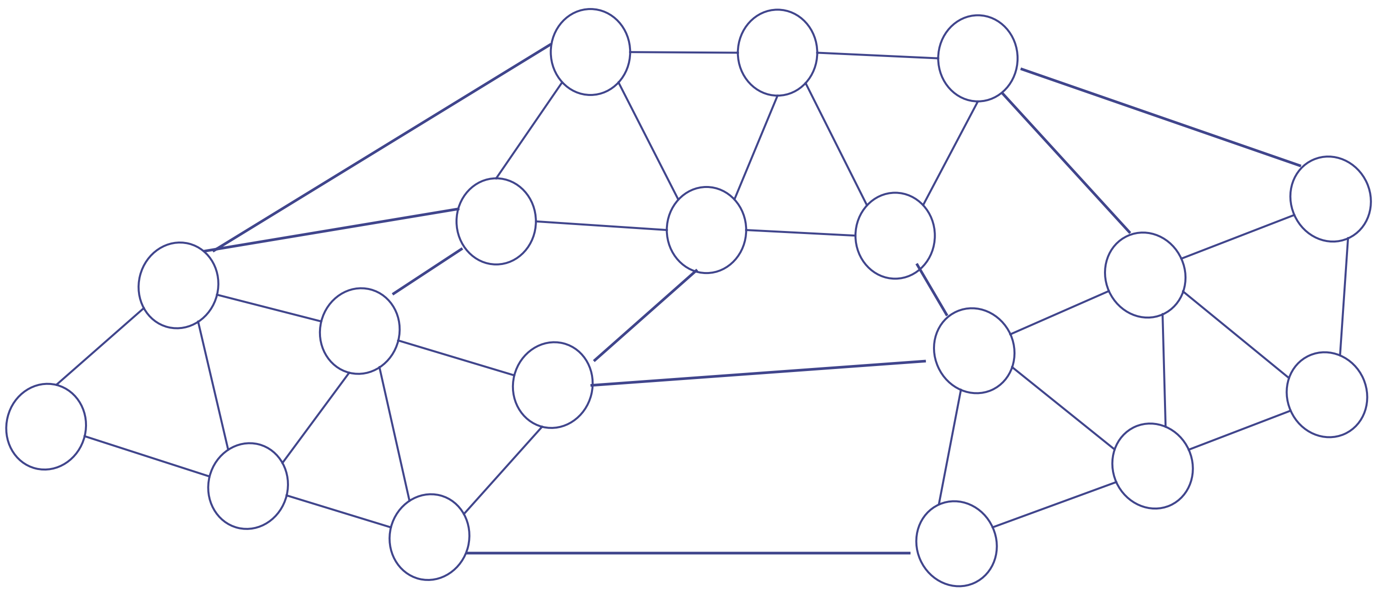
Πανεπιστήμιο Θεσσαλίας
Τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών και Δικτύων

Ασύρματα Δίκτυα Αισθητήρων

Topology & Localization



Topology Control



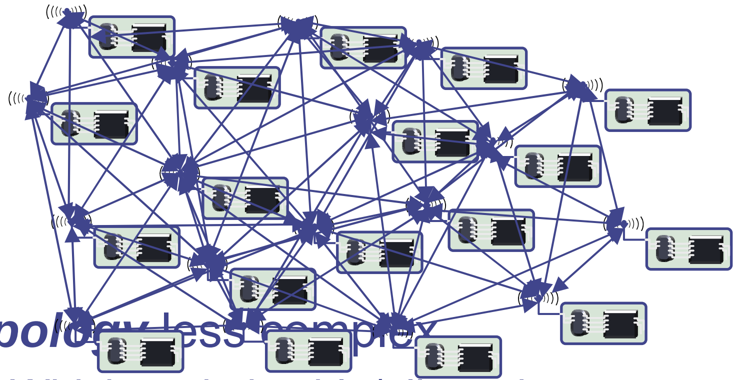
Θέματα που θα εξεταστούν

- Βασικές αρχές
- Έλεγχος κατανάλωσης ενέργειας
- Κατασκευή Backbone
- Clustering
- Adaptive node activity

Note: Presentation here follows Karl & Willig, SenSys 2003 Workshop on Wireless Sensor Networks.

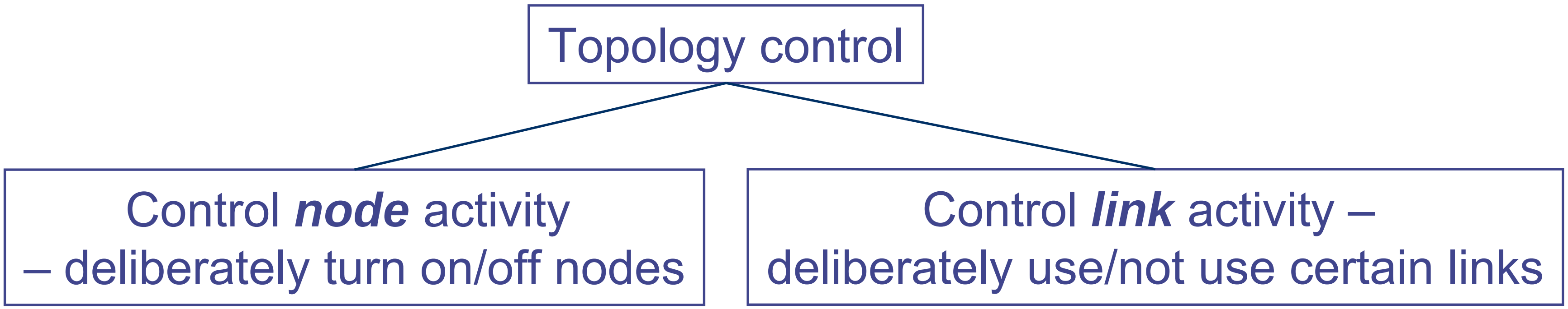
Motivation: Dense networks

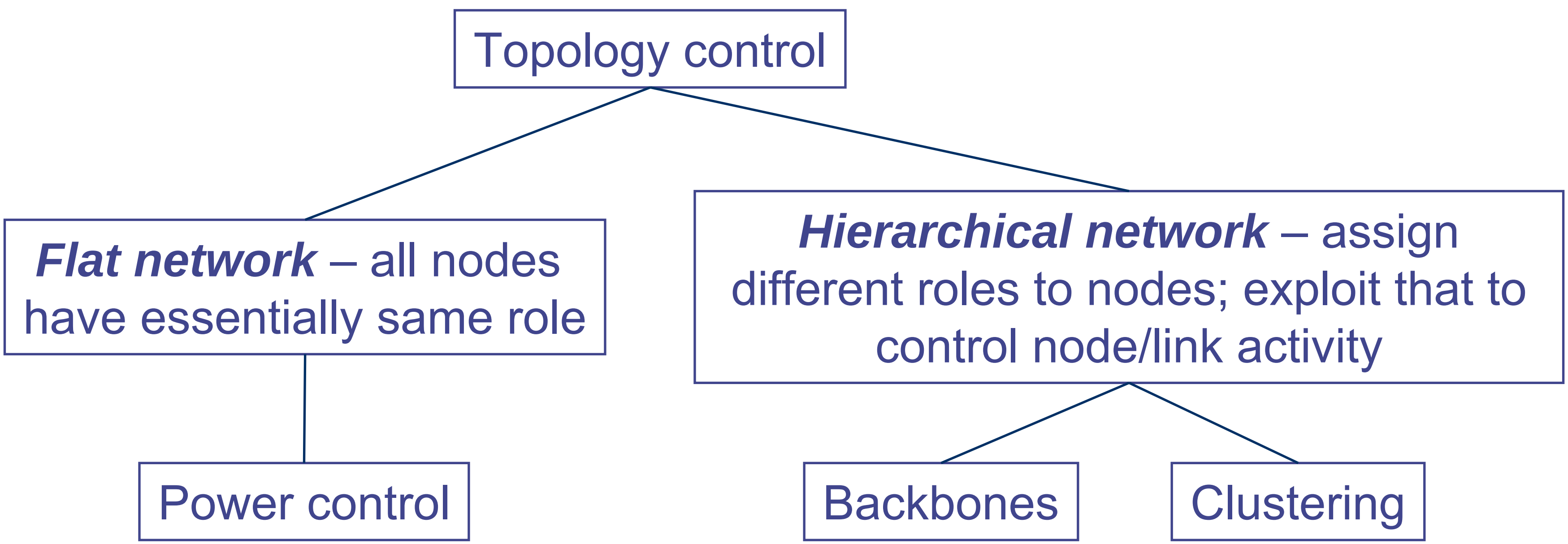
- In a very dense networks, too many nodes might be in range for an efficient operation
  - Too many collisions/too complex operation for a MAC protocol, too many paths to chose from for a routing protocol, …



- Idea: Make ***topology*** less complex
  - ***Topology***: Which node is able/allowed to communicate with which other nodes
  - Topology control needs to maintain invariants, e.g., connectivity

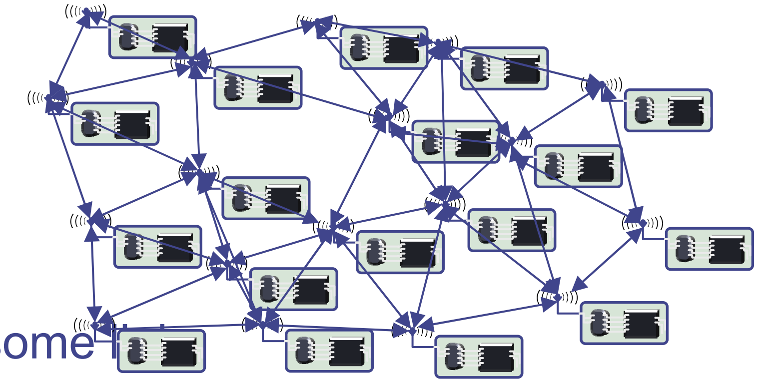
Options for topology control





Flat networks

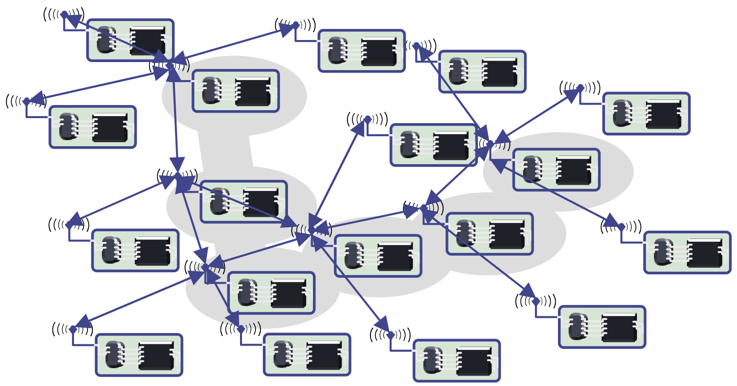
- Main option: Control transmission power
  - Do not always use maximum power
  - Selectively for some links or for a node as a whole
  - Topology looks “thinner”
  - Less interference, …



- Alternative: Selectively discard some
  - Usually done by introducing hierarchies

Hierarchical networks – backbone

- Construct a ***backbone*** network
  - Some nodes “control” their neighbors – they form a (minimal) ***dominating set***
  - Each node should have a controlling neighbor

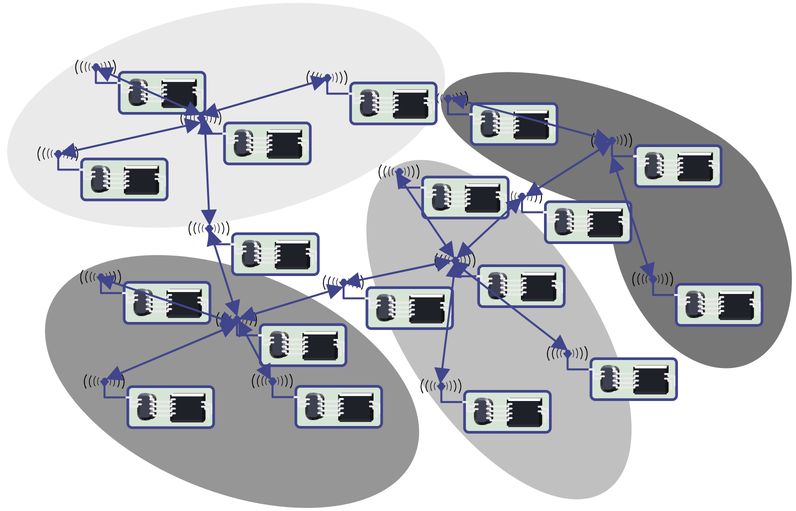


  - Controlling nodes have to be connected (backbone)
  - Only links within backbone and from backbone to controlled neighbors are used
- Formally: Given graph G=(V,E), construct D ½ V such that

$$\forall v \in V : v \in D \vee \exists d \in D : (v,d) \in E$$

Hierarchical network – clustering

- Construct ***clusters***
  - Partition nodes into groups ("clusters")
  - Each node in exactly one group
    - Except for nodes "bridging" between two or more groups
  - Groups can have ***clusterheads***



  - Typically: all nodes in a cluster are direct neighbors of their clusterhead
  - Clusterheads are also a dominating set, but should be separated from each other – they form an ***independent set***
- Formally: Given graph G=(V,E), construct C ½ V such that

$$\forall v \in V - C : \exists c \in C : (v, c) \in E$$

$$\forall c_1, c_2 \in C : (c_1, c_2) \notin E$$

Power control – magic numbers?

- Question: What is a good power level for a node to ensure “nice” properties of the resulting graph?
- Idea: Controlling transmission power corresponds to controlling the number of neighbors for a given node
- Is there an “optimal” number of neighbors a node should have?
  - Is there a “magic number” that is good irrespective of the actual graph/network under consideration?
- Historically, k=6 or k=8 had been suggested as such “magic numbers”
  - However, they optimize progress per hop – they do ***not*** guarantee connectivity of the graph!!

! Needs deeper analysis

Controlling transmission range

- Assume all nodes have identical transmission range r=r(|V|), network covers area A, V nodes, uniformly distr.
- Fact: Probability of connectivity goes to zero if:
- Fact: $r(|V|) \leq \sqrt{\frac{(1-\epsilon)A \log |V|}{\pi |V|}}$, for any $\epsilon > 0$

if and only i

- Fact (unifor $r(|V|) \geq \sqrt{\frac{A(\log |V| + \gamma_{|V|})}{\pi |V|}}$

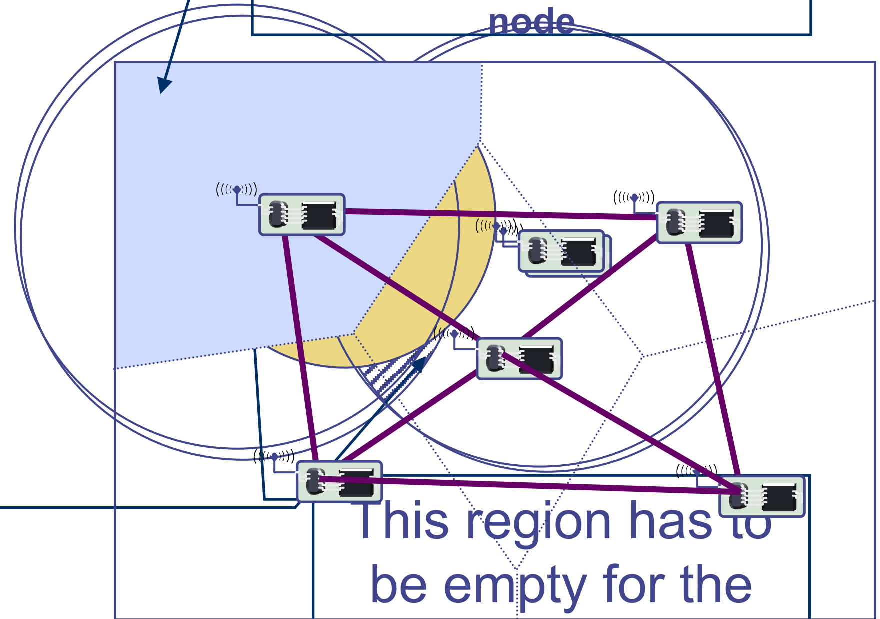
$$P(\text{G is } k\text{-connected}) \approx \left(1 - \sum_{l=0}^{k-1} \frac{(\rho \pi r^2)^l}{l!} e^{-\rho \pi r^2}\right)$$

Controlling number of neighbors

- Knowledge about range also tells about number of neighbors
  - Assuming node distribution (and density) is known, e.g., uniform
- Alternative: directly analyze number of neighbors
  - Assumption: Nodes randomly, uniformly placed, only transmission range is controlled, identical for all nodes, only symmetric links are considered
- Result: For connected network, required number of neighbors per node is **Θ (log |V|)**
  - It is ***not a constant***, but depends on the number of nodes!
  - For a larger network, nodes need to have more neighbors & larger transmission range! – Rather inconvenient
  - Constants can be bounded

Some example constructions for power control

- Basic idea for most of the following methods:
  Take a graph G=(V,E), produce a graph $G^0=(V,E^0)$ that maintains connectivity with fewer edges
  - Relative Neighborhood Graph (RNG)
  - Gabriel graph
  - Delaunay triangulation

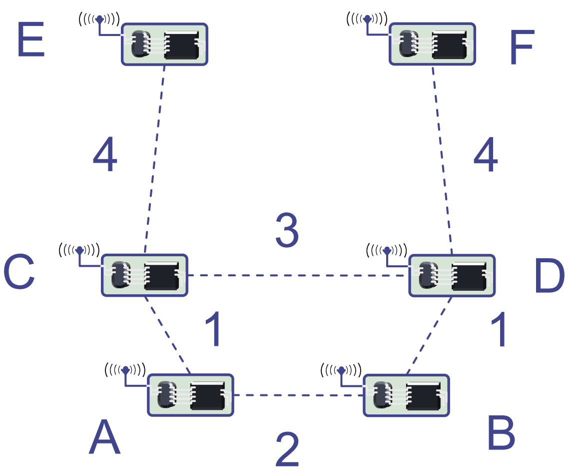


Centralized power control algorithm

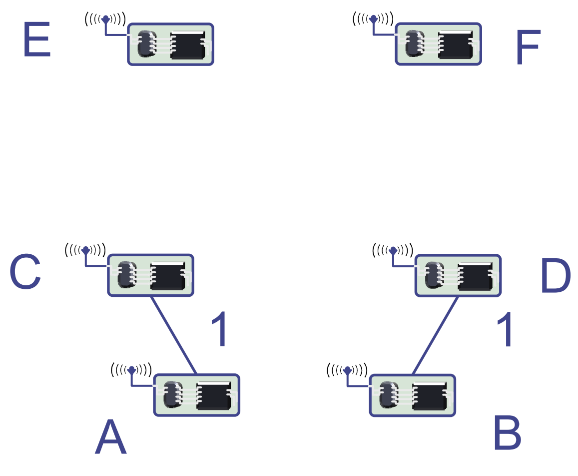
- Goal: Find topology control algorithm minimizing the ***maximum*** power used by any node
  - Ensuring simple or bi-connectivity
  - Assumptions: Locations of all nodes and path loss between all node pairs are known; each node uses an individually set power level to communicate with all its neighbors
- Idea: Use a centralized, greedy algorithm
  - Initially, all nodes have transmission power 0
  - Connect those two components with the shortest distance between them (raise transmission power accordingly)
- Second phase: Remove links (=reduce transmission power) not needed for connectivity
- Exercise: Relation to Kruskal's MST algorithm?

Centralized power control algorithm

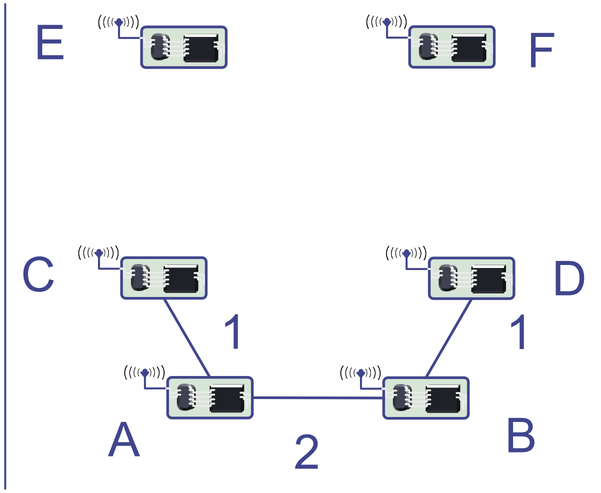
Topology



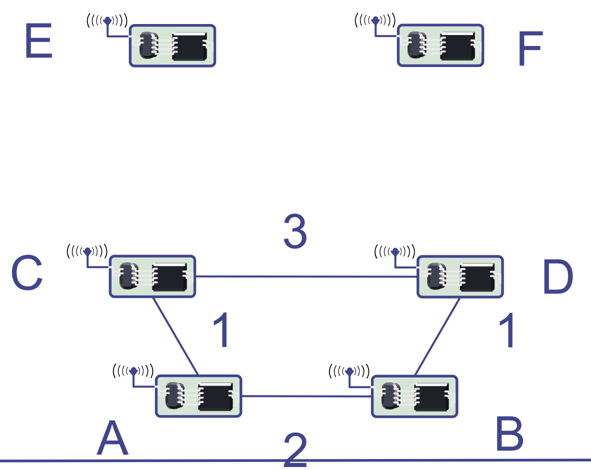
1) Connect A-C and B-D



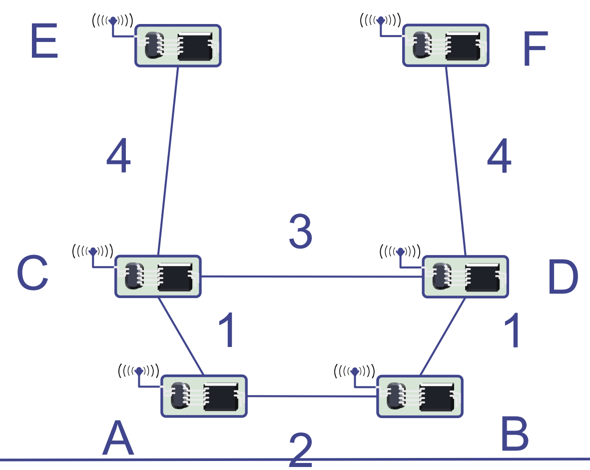
2) Connect A-B



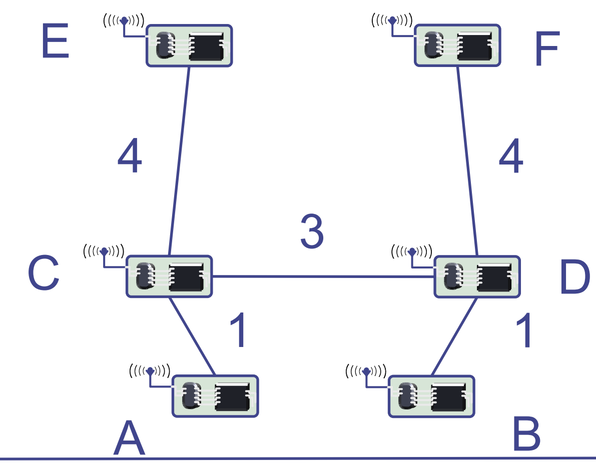
3) Connect C-D



4) Connect C-E and D-F



5) Remove edge A-B



Overview

- Motivation, basics
- Power control
- ***Backbone construction***
- Clustering
- Adaptive node activity

Hierarchical networks – backbones

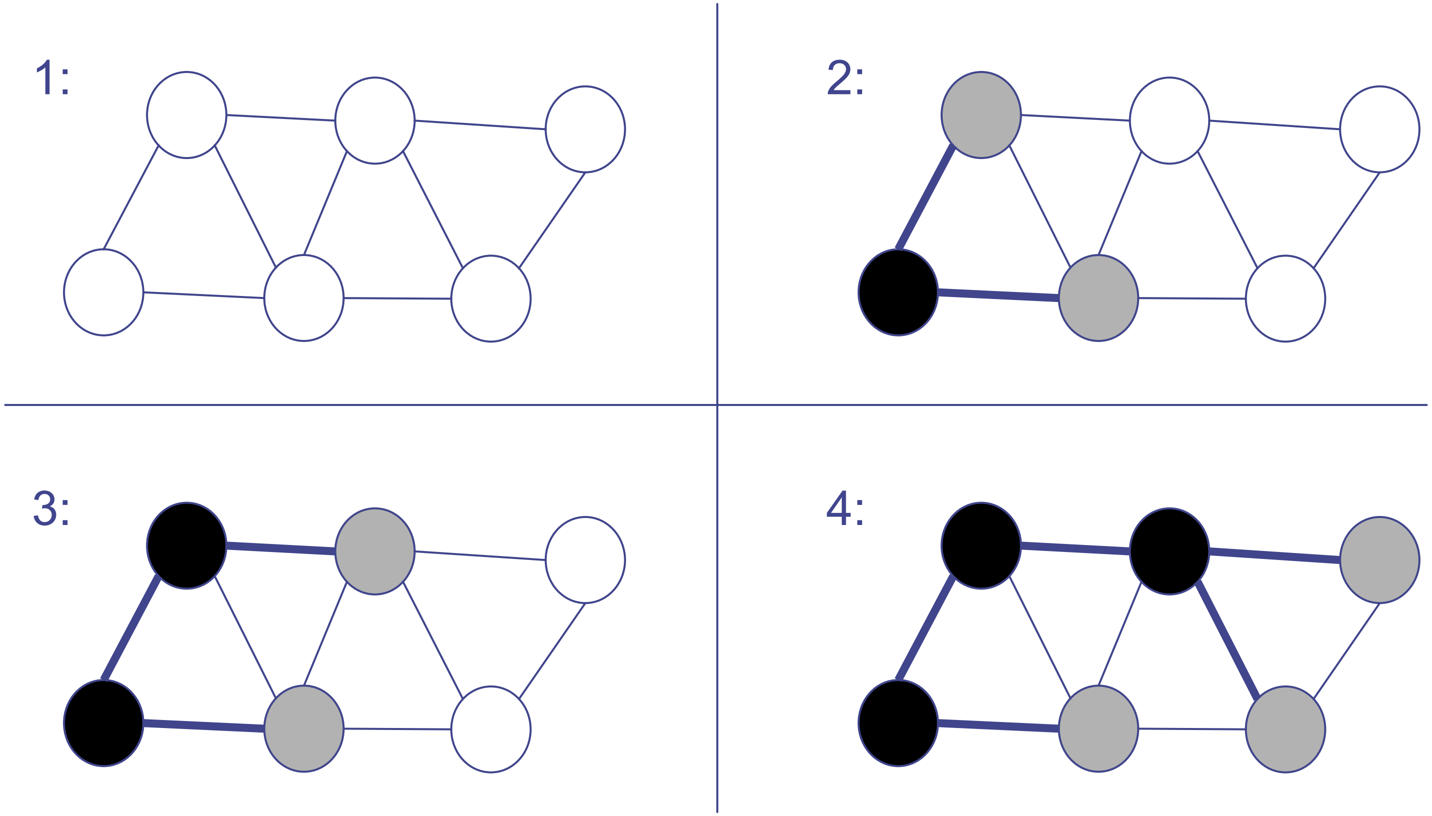
- Idea: Select some nodes from the network/graph to form a ***backbone***
  - A connected, minimal, dominating set (MDS or MCDS)
  - Dominating nodes control their neighbors
  - Protocols like routing are confronted with a simple topology – from a simple node, route to the backbone, routing in backbone is simple (few nodes)
- Problem: MDS is an NP-hard problem
  - Hard to approximate, and even approximations need quite a few messages

Backbone by growing a tree

- Construct the backbone as a tree, grown iteratively

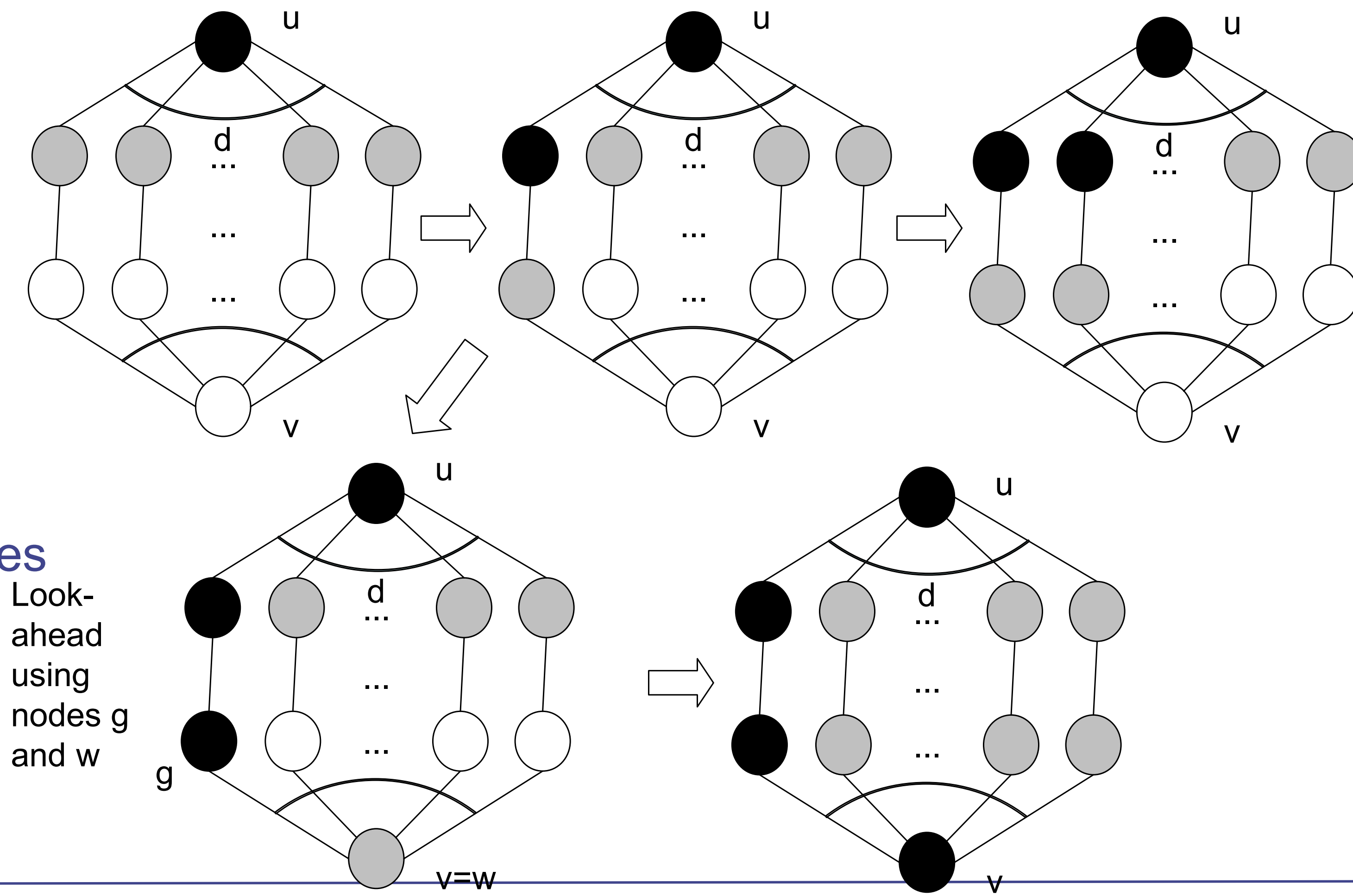
```
initialize all nodes' color to white
pick an arbitrary node and color it grey

while (there are white nodes) {
  pick a grey node v that has white neighbors
  color the grey node v black
  foreach white neighbor u of v {
    color u grey
    add (v,u) to tree T
  }
}
```

Backbone by growing a tree – Example



Problem: Which gray node to pick?

- When blindly picking any gray node to turn black, resulting tree can be very bad



Solution:
Look ahead!
One step suffices

Performance of tree growing with look ahead

- Dominating set obtained by growing a tree with the look ahead heuristic is at most a factor $2(1+ H(\Delta))$ larger than MDS
  - $H(¢)$ harmonic function, $H(k) = \sum_{i=1}^{k} 1/i <= \ln k + 1$
  - Δ is maximum degree of the graph
- It is automatically connected
- Can be implemented in a distributed fashion as well

Start big, make lean

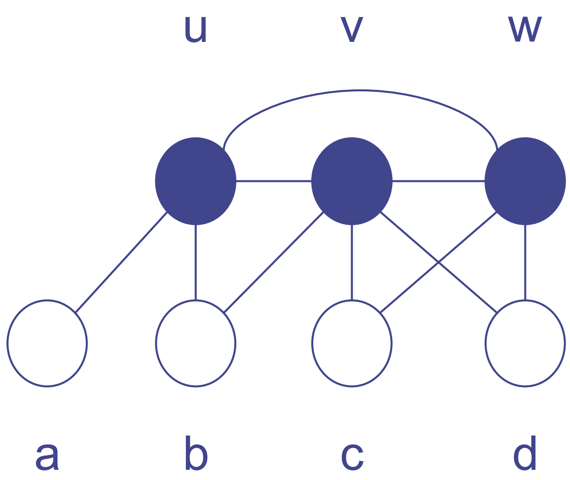
- Idea: start with some, possibly large, connected dominating set, reduce it by removing unnecessary nodes
- Initial construction for dominating set
  - All nodes are initially white
  - Mark any node black that has two neighbors that are not neighbors of each other (they might need to be dominated)

  ! Black nodes form a connected dominating set (proof by contradiction); shortest path between ANY two nodes only contains black nodes

- Needed: Pruning heuristics

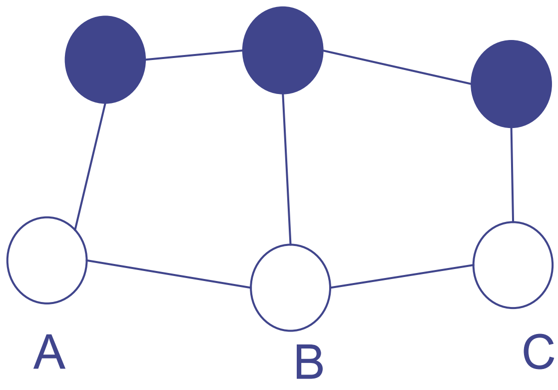
Pruning heuristics

- Heuristic 1: Unmark node v if
  - Node v and its neighborhood are included in the neighborhood of some node marked node u (then u will do the domination for v as well)
  - Node v has a smaller unique identifier than u (to break ties)
- Heuristic 2: Unmark node v if
  - Node v's neighborhood is included in the neighborhood of two marked neighbors u and w
  - Node v has the smallest identifier of the tree nodes
- Nice and easy, but only linear approximation factor



One more distributed backbone heuristic: Span

- Construct backbone, but take into account need to carry traffic – preserve capacity
  - Means: If two paths could operate without interference in the original graph, they should be present in the reduced graph as well
  - Idea: If the stretch factor (induced by the backbone) becomes too large, more nodes are needed in the backbone
- Rule: Each node observes traffic around itself
  - If node detects two neighbors that need three hops to communicate with each other, node joins the backbone, shortening the path
  - Contention among potential new backbone nodes handled using random backoff



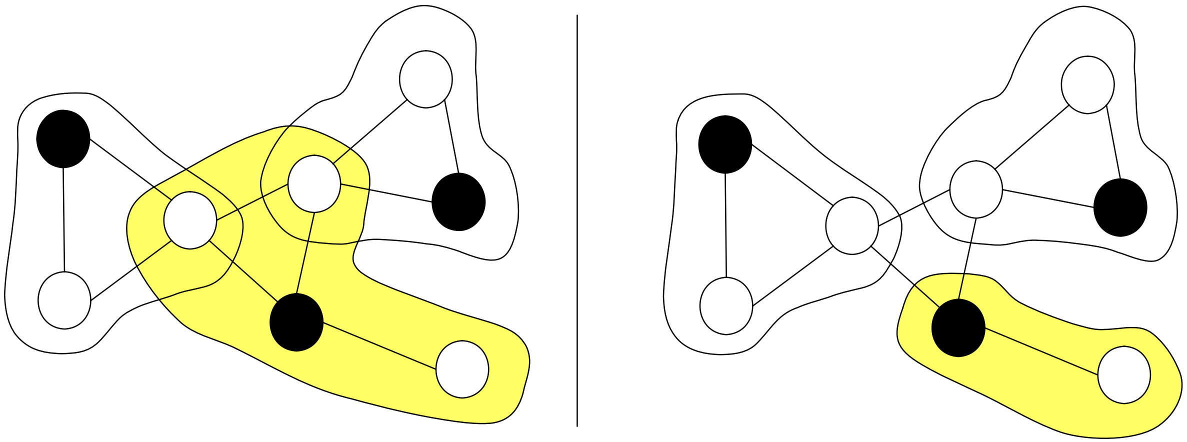
Overview

- Motivation, basics
- Power control
- Backbone construction
- ***Clustering***
- Adaptive node activity

Clustering

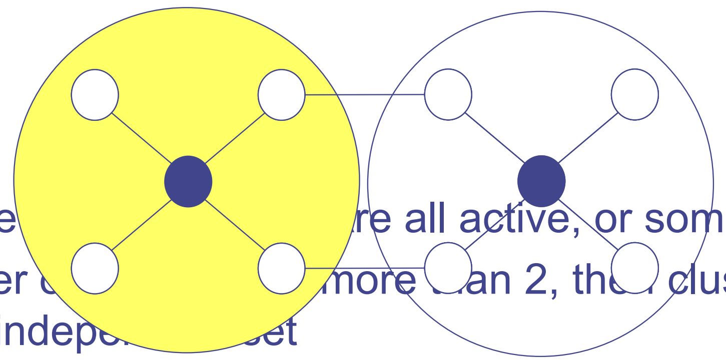
- Partition nodes into groups of nodes – ***clusters***
- Many options for details
  - Are there ***clusterheads***? – One controller/representative node per cluster
  - May clusterheads be neighbors? If no: clusterheads form an ***independent set C:*** Typically: clusterheads form a ***maximum independent set***
  - May clusters overlap? Do they have nodes in common?

$$\forall c_1, c_2 \in C : (c_1, c_2) \notin E$$



Clustering

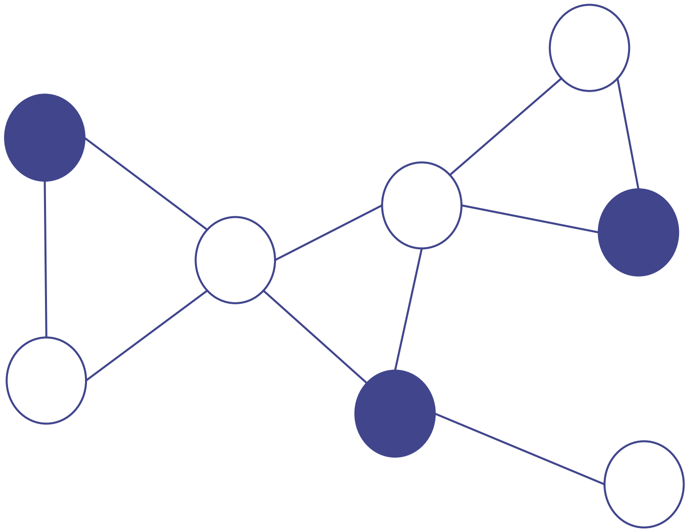
- Further options
  - How do clusters communicate? Some nodes need to act as ***gateways*** between clusters
    If clusters may not overlap, two nodes need to jointly act as a ***distributed gateway***



  - How many gateways exist be[illegible]re all active, or some standby?
  - What is the maximal diameter [illegible]more [illegible]an 2, the[illegible] clusterheads are not necessarily a maximum indepe[illegible]et
  - Is there a hierarchy of clusters?

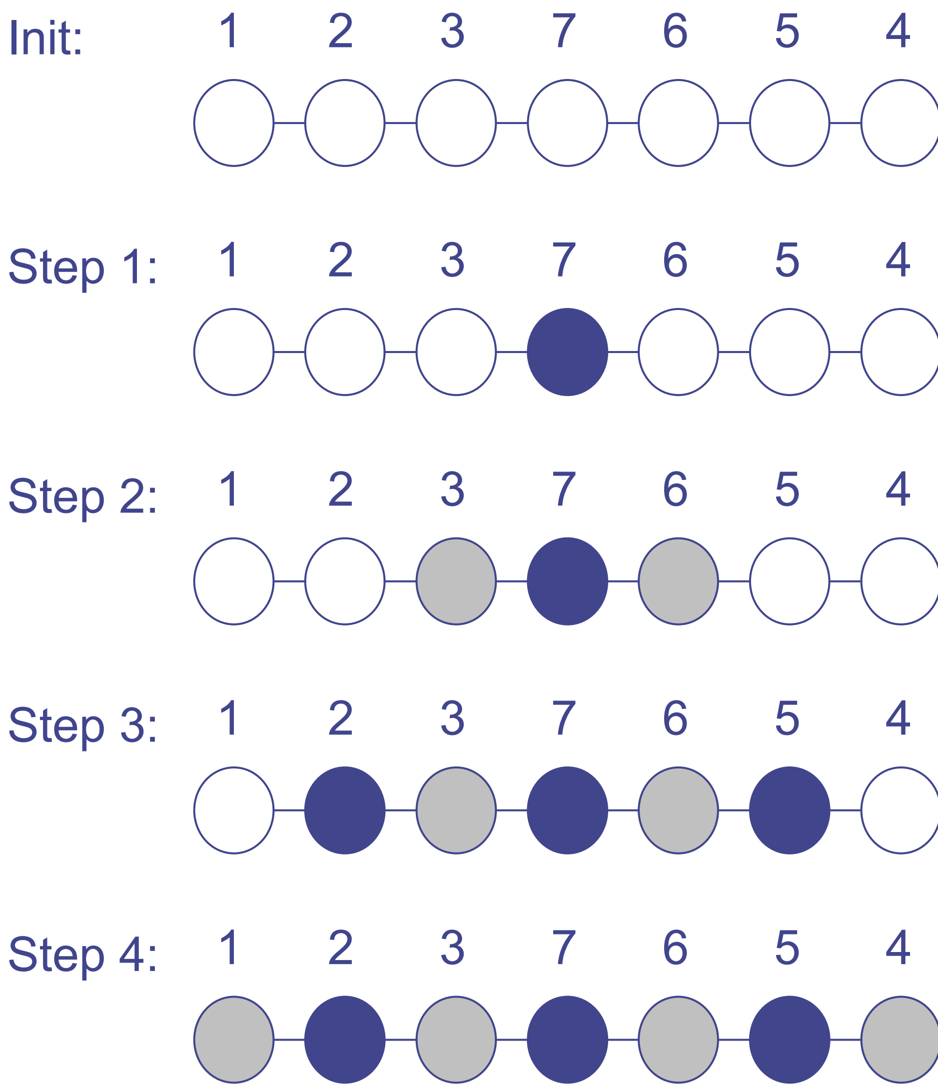
Maximum independent set

- Computing a maximum independent set is NP-complete
- Can be approximate within $(\Delta + 3)/5$ for small Δ, within $O(\Delta \log \log \Delta / \log \Delta)$ else; Δ bounded degree
- Show: A maximum independent set is also a dominating set
- Maximum independent set not necessarily intuitively desired solution
  - Example: Radial graph, with only (v_0,v_i) 2 E



A basic construction idea for independent sets

- Use some attribute of nodes to break local symmetries
  - Node identifiers, energy reserve, mobility, weighted combinations… - matters not for the idea as such (all types of variations have been looked at)
- Make each node a clusterhead that locally has the largest attribute value
- Once a node is dominated by a clusterhead, it abstains from local competition, giving other nodes a chance



Determining gateways to connect clusters

- Suppose: Clusterheads have been found
- How to connect the clusters, how to select gateways?

- It suffices for each clusterhead to connect to all other clusterheads that are at most three hops
  - Resulting backbone (!) is connected

- Formally: Steiner tree problem
  - Given: Graph G=(V,E), a subset C ½ V
  - Required: Find another subset T ½ V such that S [T is connected and S [T is a cheapest such set
  - Cost metric: number of nodes in T, link cost
  - Here: special case since C are an independent set

Rotating clusterheads

- Serving as a clusterhead can put additional burdens on a node
  - For MAC coordination, routing, …
- Let this duty rotate among various members
  - Periodically reelect – useful when energy reserves are used as discriminating attribute
  - LEACH – determine an optimal percentage P of nodes to become clusterheads in a network
    - Use 1/P rounds to form a period
    - In each round, nP nodes are elected as clusterheads
    - At beginning of round r, node that has not served as clusterhead in this period becomes clusterhead with probability P/(1-p(r mod 1/P))

Multi-hop clusters

- Clusters with diameters larger than 2 can be useful, e.g., when used for routing protocol support
- Formally: Extend "domination" definition to also dominate nodes that are at most d hops away
- Goal: Find a smallest set D of dominating nodes with this extended definition of dominance
- Only somewhat complicated heuristics exist

- Different tilt: Fix the ***size*** (not the diameter) of clusters
  - Idea: Use ***growth budgets*** – amount of nodes that can still be adopted into a cluster, pass this number along with broadcast adoption messages, reduce budget as new nodes are found

Passive clustering

- Constructing a clustering structure brings overheads
  - Not clear whether they can be amortized via improved efficiency
- Question: Eat cake and have it?
  - Have a clustering structure without any overhead?
  - Maybe not the best structure, and maybe not immediately, but benefits at zero cost are no bad deal…

! Passive clustering
  - Whenever a broadcast message travels the network, use it to construct clusters on the fly
  - Node to start a broadcast: Initial node
  - Nodes to forward this first packet: Clusterhead
  - Nodes forwarding packets from clusterheads: ordinary/gateway nodes
  - And so on… ! Clusters will emerge at low overhead

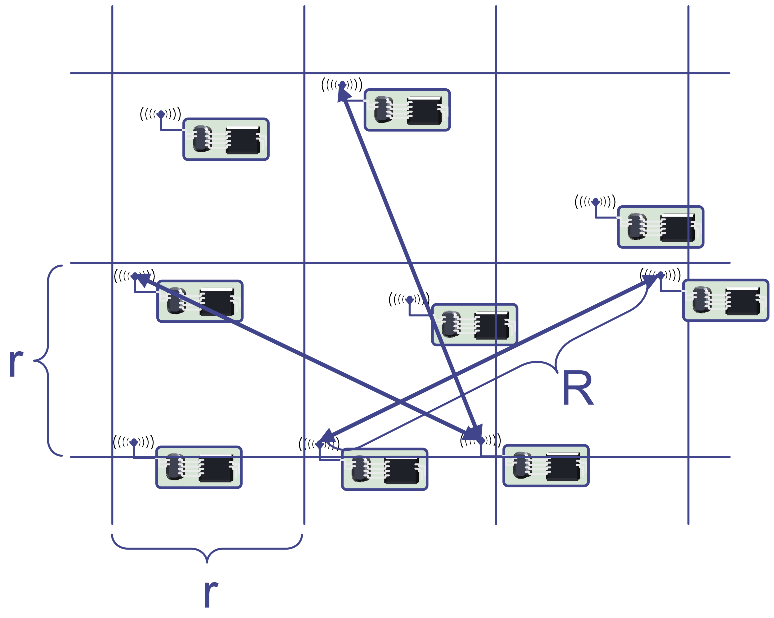
Overview

- Motivation, basics
- Power control
- Backbone construction
- Clustering
- ***Adaptive node activity***

Adaptive node activity

- Remaining option: Turn some nodes off deliberately
- Only possible if other nodes remain on that can take over their duties
- Example duty: Packet forwarding
  - Approach: Geographic Adaptive Fidelity (GAF)

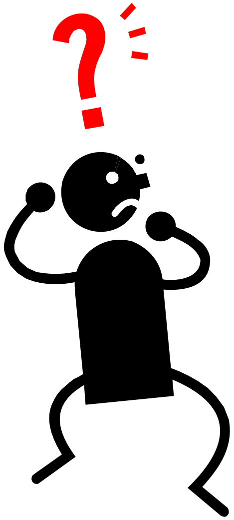
- Observation: Any two nodes within a square of length $r < R/5^{1/2}$ can replace each other with respect to forwarding
  - R radio range
- Keep only one such node active, let the other sleep



Conclusion

- Various approaches exist to trim the topology of a network to a desired shape
- Most of them bear some non-negligible overhead
  - At least: Some distributed coordination among neighbors, or they require additional information
  - Constructed structures can turn out to be somewhat brittle – overhead might be wasted or even counter-productive
- Benefits have to be carefully weighted against risks for the particular scenario at hand

Ερωτήσεις;



Overview

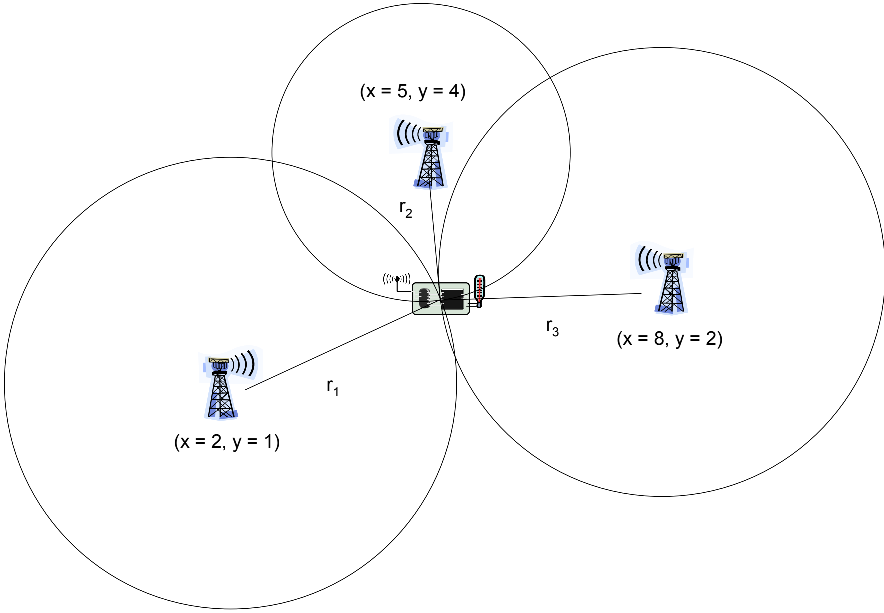
- ***Basic approaches***
- Trilateration
- Multihop schemes

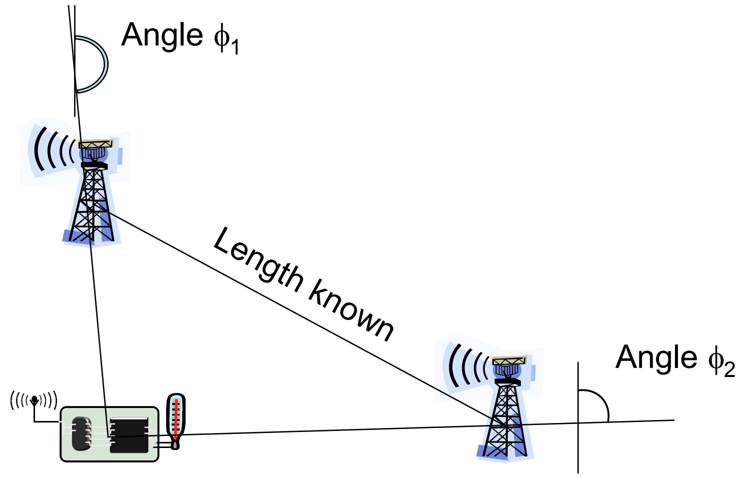
Localization & positioning

- Determine ***physical position*** or ***logical location***
  - Coordinate system or symbolic reference
  - Absolute or relative coordinates
- Options
  - Centralized or distributed computation
  - Scale (indoors, outdoors, global, …)
  - Sources of information
- 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, …

Main approaches (information sources)

- 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 analysis
  - Radio environment has characteristic “signatures”
  - Can be measured beforehand, stored, compared with current situation



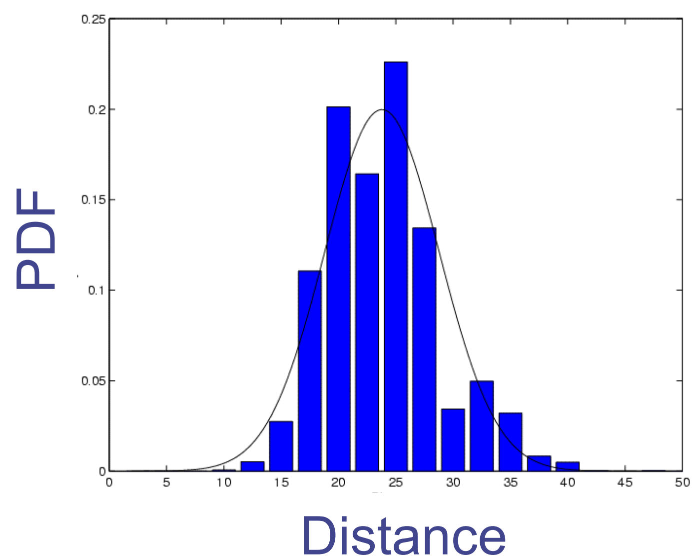


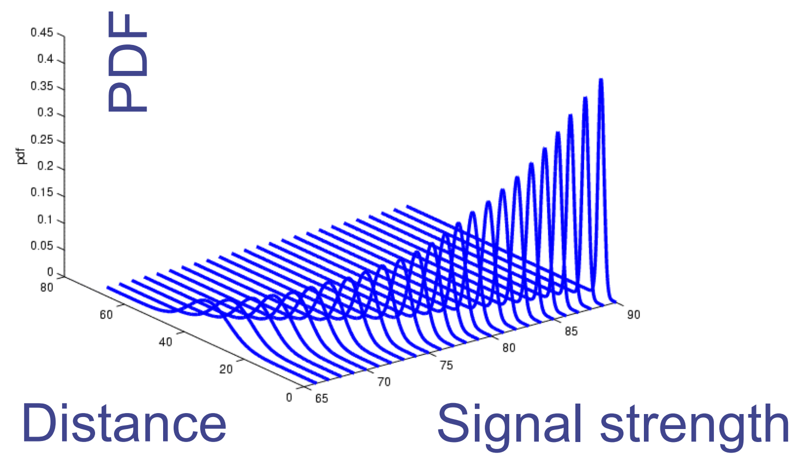
Estimating distances – RSSI

- Received Signal Strength Indicator
  - Send out signal of known strength, use received signal strength and path loss coefficient to estimate distance

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

  - Problem: … a fixed RSSI





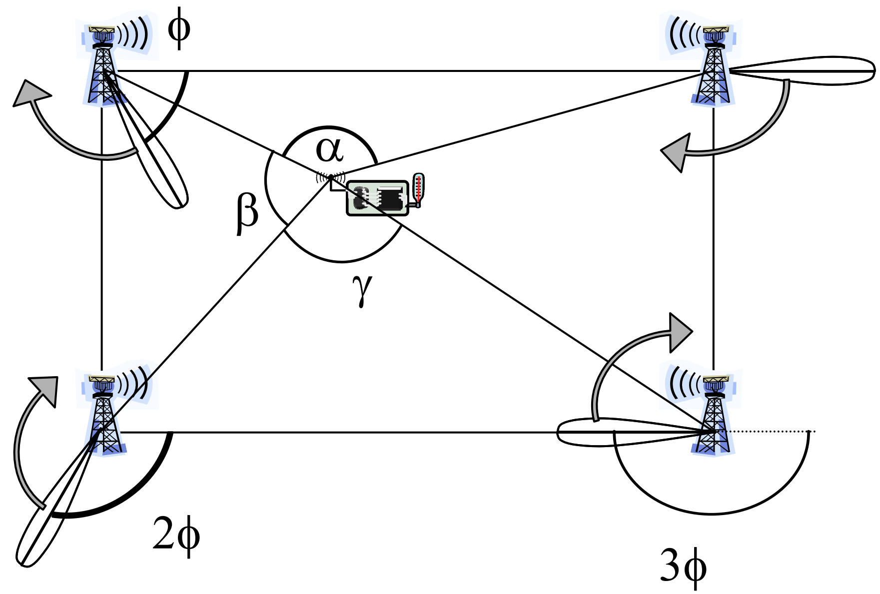
Estimating distances – other means

- Time of arrival (ToA)
  - Use time of transmission, propagation speed, time of arrival to compute distance
  - Problem: Exact time synchronization

- 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
  - Compute difference between arrival times to compute distance
  - Problem: Calibration, expensive/energy-intensive hardware

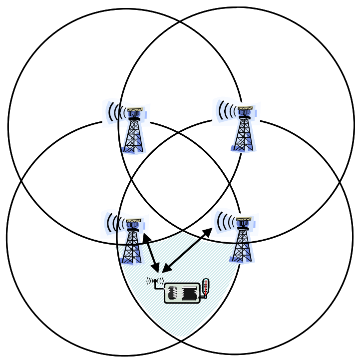
Determining angles

- 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

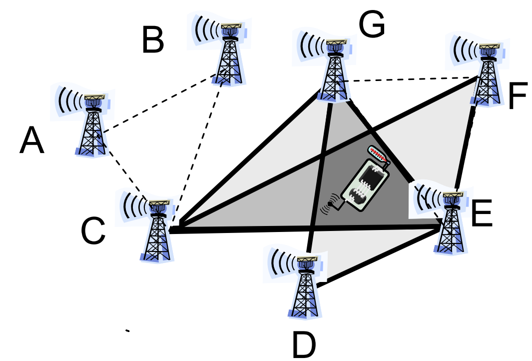


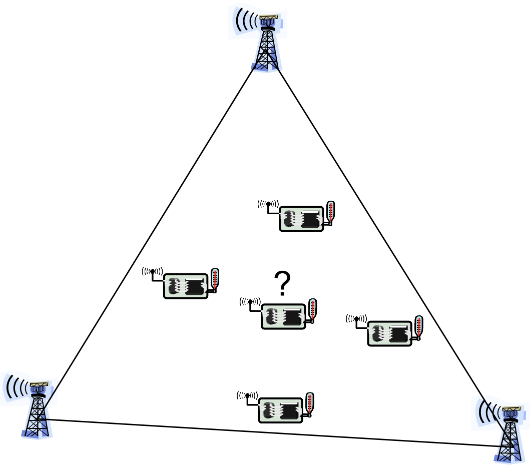
Some range-free, single-hop localization techniques

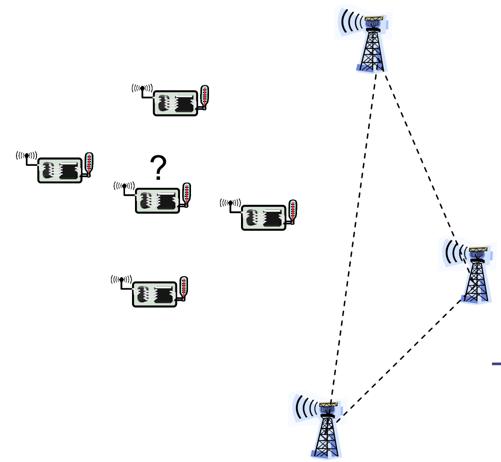
- ***Overlapping connectivity***: Position is estimated in the center of area where circles from which signal is heard/not heard overlap



- ***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







Overview

- Basic approaches
- ***Trilateration***
- Multihop schemes

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
  - Subtracting eq. 3 from 1 & 2:

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

-
$$(x_1 - x_u)^2 - (x_3 - x_u)^2 + (y_1 - y_u)^2 - (y_3 - y_u)^2 = r_1^2 - r_3^2$$
$$(x_2 - x_u)^2 - (x_2 - x_u)^2 + (y_2 - y_u)^2 - (y_2 - y_u)^2 = r_2^2 - r_3^2.$$

$$2(x_3 - x_1)x_u + 2(y_3 - y_1)y_u = (r_1^2 - r_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2)$$
$$2(x_3 - x_2)x_u + 2(y_3 - y_2)y_u = (r_2^2 - r_2^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2)$$

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

!

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

Trilateration with distance errors

- What if only distance estimation $r_i^0 = r_i + \varepsilon_i$ available?
- Use multiple anchors, overdetermined system of equations
- $$2\begin{bmatrix} x_n - x_1 & y_n - y_1 \\ \vdots & \vdots \\ x_n - x_{n-1} & y_n - y_{n-1} \end{bmatrix} \begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} (r_1^2 - r_n^2) - (x_1^2 - x_n^2) - (y_1^2 - y_n^2) \\ \vdots \\ (r_{n-1}^2 - r_n^2) - (x_{n-1}^2 - x_n^2) - (y_{n-1}^2 - y_n^2) \end{bmatrix}$$

$$||\mathbf{Ax} - \mathbf{b}||_2$$

Minimize mean square error

- Look at square of the of Euclidean norm expression (note that for all vectors
  $$||\mathbf{v}||_2^2 = \mathbf{v}^T\mathbf{v}$$

$$||\mathbf{Ax} - \mathbf{b}||_2^2 = (\mathbf{Ax} - \mathbf{b})^T(\mathbf{Ax} - \mathbf{b}) = \mathbf{x}^T\mathbf{A}^T\mathbf{Ax} - 2\mathbf{x}^T\mathbf{A}^T\mathbf{b} + \mathbf{b}^T\mathbf{b}$$

-

$$2\mathbf{A}^T\mathbf{Ax} - 2\mathbf{A}^T\mathbf{b} = 0 \Leftrightarrow \mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$$

- ean square error

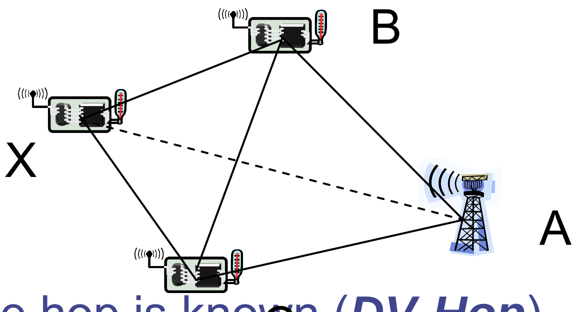
- Essentially similar for angulation as well

Overview

- Basic approaches
- Trilateration
- ***Multihop schemes***

Multihop range estimation

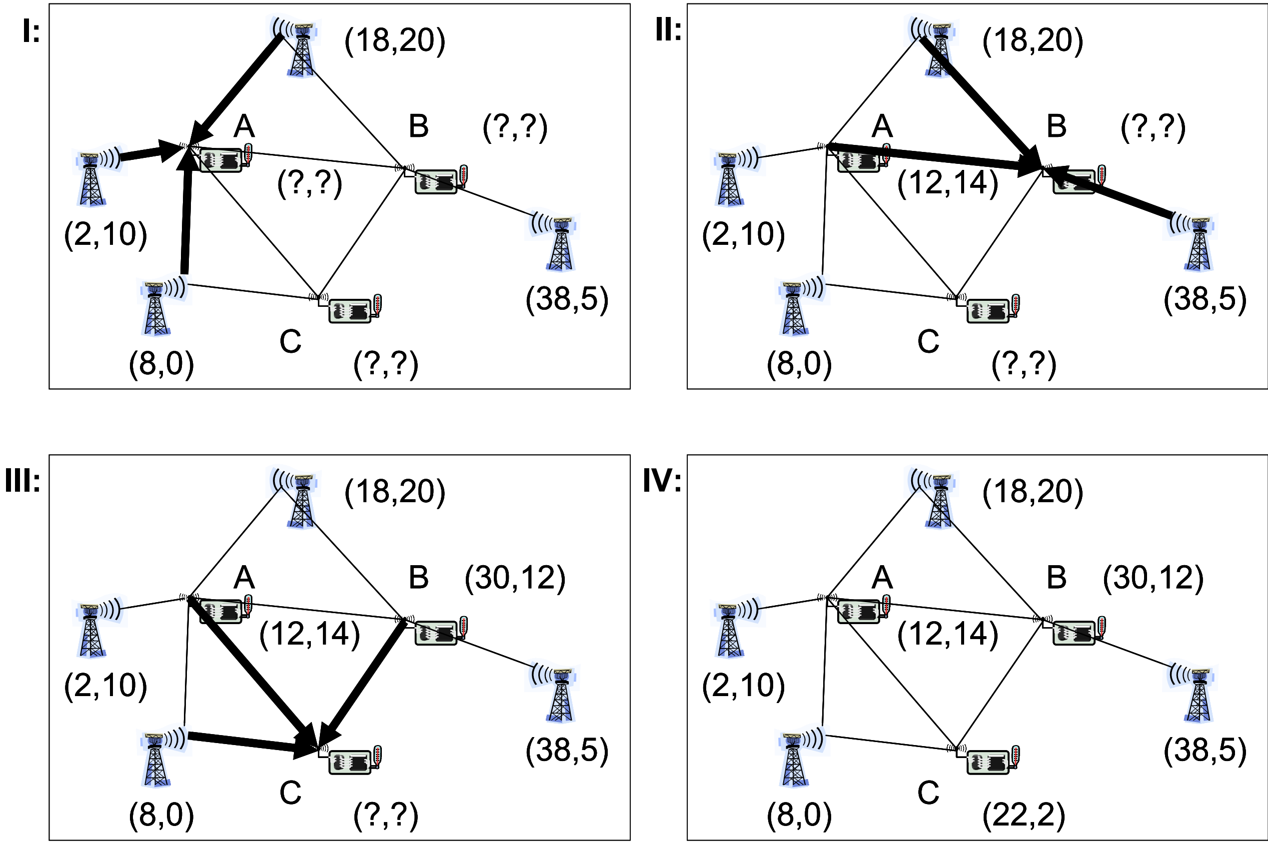
- How to estimate range to a node to which no direct radio communication exists?
  - No RSSI, TDoA, …
  - But: Multihop communication is possible



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

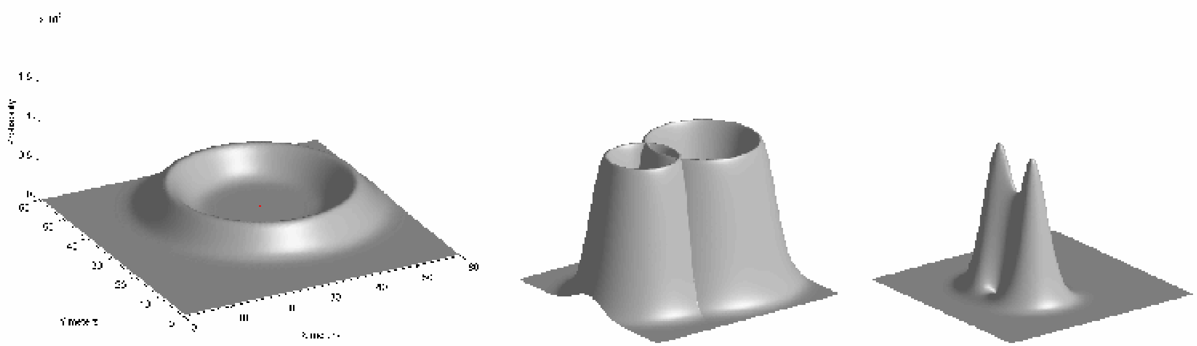
Iterative multilateration

- 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



Probabilistic position description

- Similar idea to previous one, but accept problem that position of nodes is only probabilistically known
  - Represent this probability explicitly, use it to compute probabilities for further nodes



(a) Probability density function of a node positions after receiving a distance estimate from one anchor

(b) Probability density functions of two distance measurements from two independent anchors

(c) Probability density function of a node after intersecting two anchor's distance measurements

Conclusions

- Determining location or position is a vitally important function in WSN, but fraught with many errors and shortcomings
  - Range estimates often not sufficiently accurate
  - Many anchors are needed for acceptable results
  - Anchors might need external position sources (GPS)
  - Multilateration problematic (convergence, accuracy)

Ερωτήσεις;