CS691AA: Challenges in Node Deployment and Topology Control

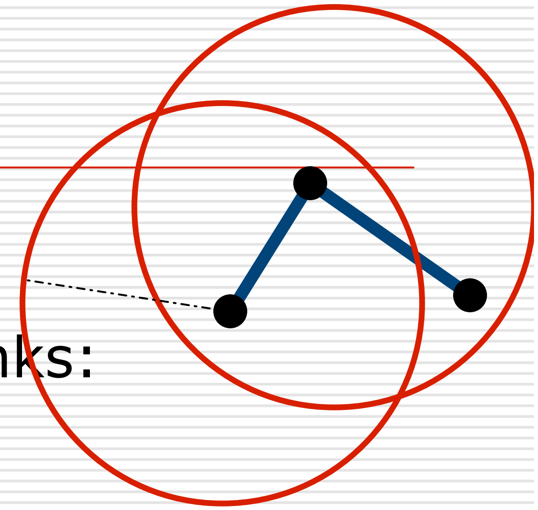
Deepak Ganesan
Assistant Professor,
Computer Science Department
http://www.cs.umass.edu/~dganesan/
http://www.cs.umass.edu/~dganesan/courses/fall04/
Office Hours: Mon 10-11am, 250 CS Building
Mailing List: cs691aa@cs.umass.edu

Deployment Configuration and Topology Control

- Deployment Questions
  - Case 1: Random Placement (dropped, scattered)
    - What are the properties of connectivity and coverage in random graphs?
  - Case 2: Structured Placement
    - Where and how to place nodes while ensuring that connectivity and coverage is preserved while power is minimized?
- Topology Control Questions
  - Exploiting deployment density:
    - Given a dense deployment where not all nodes are required to create a connected topology, how to duty-cycle nodes to save power?
  - Methodology:
    - Empirical adaptation (assume nothing about initial placement)
    - Analytical adaptation (assuming given density/placement)

Randomly Placed Networks

Network Connectivity



- Idealized* geometric model for wireless links: perfect connectivity within radio range R
- Network graph G formed by nodes as vertices and these links as edges
- Basic notion of connectivity: there exists at least one multihop path between any pair of nodes in the network, could be generalized to k-connectivity, existence of Hamiltonian cycle etc.

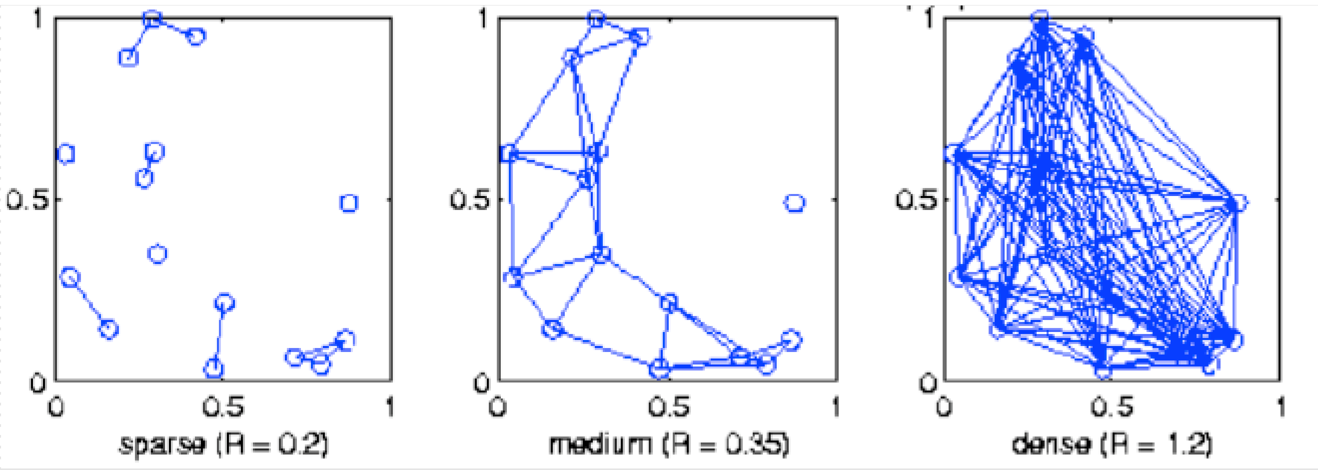
***Caveat: Perhaps good for preliminary analysis, but known to be unrealistic**

Random Deployment

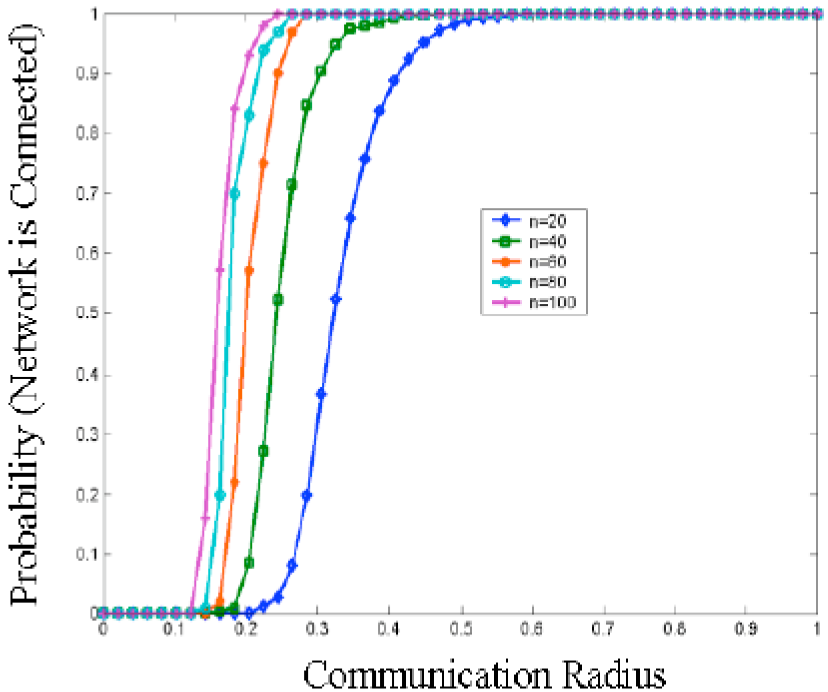
- E.g. scattered from an aircraft/robot, mixed into concrete
- Issues of average density and range settings are important
- Connectivity issues can be explored using the Theory of Random Graphs and Percolation Theory

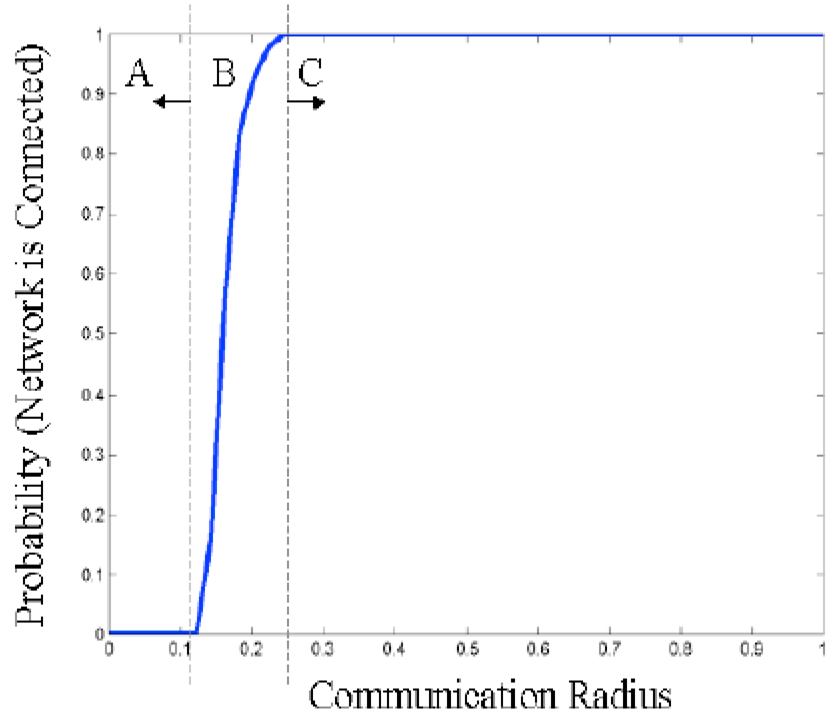
Random Graphs

- Bernoulli Random Graphs G(n,p): edge between any pair of the n nodes independently with probability p
- Geometric Random Graphs G(n,R): n nodes placed with a uniform random distribution in a finite region; edge between any pair of nodes within range R.



Phase Transition in Connectivity





- Gupta-Kumar '98 have shown that asymptotically O(log n) neighbors are required on average for each node for network to be connected
- Edge of transition is power-efficient/minimum-cost

Phase Transition in Connectivity

- 2D model of Gupta & Kumar (1998) involves laborious calculations, results from continuum percolation theory.
- Simpler 1D Model: Poisson arrivals with average of λ node per unit length. Connectivity between nodes within range R.

- Probability that first n nodes form a connected network:

$$P_n(R) = \prod_{i=1}^{n-1} (1 - e^{-\lambda R}) = (1 - e^{-\lambda R})^{(n-1)}$$

- This model also shows an analogous phase transition with an O(log n) density threshold function (just set R = log n / λ).

Many other properties (k-connectivity, partition into cliques, existence of Hamiltonian cycle, k-coloring etc.) undergo phase transitions

Impact of Phase Transitions

- If the goal is to construct a connected network in a dense deployment, only a small number of backbone nodes needs to be active (not asleep).

- Need to ensure that sufficient density of nodes is active at any given time to ensure a connected topology.

Why do we need dense sensor networks?

- Over-deployment at deployment time can help increase robustness to failure and extend lifetime
  - Not always possible to do additional deployment on-demand(e.g. emergency services).
  - Expensive to deploy on-demand (when nodes die)
- Over-deployment may be necessary to achieve required sensing granularity.
  - Sensors are more effective when they are in close proximity to the phenomenon (~ r^{-2} -- r^{-4})
  - Multiple sensing points can provide greater capability (sensor arrays such as in radar systems or binocular vision).
- Over-deployment may not be essential if there are good power harvesting schemes.
  - Alternative non-wired power sources (solar panel, micro-engines, etc.), but not always practical or available.

How can we exploit node density to save power?

Topology Control Techniques

- Core Idea
  - Find a subset (k) of nodes that provide communication coverage.
  - Cycle through nodes such that at least/only k of them are active at any time and the rest are asleep.
- Minimal Connected Dominating Set (MCDS)
  - Find the dominating set, and then find a subset of nodes that connect all the nodes in the dominating set.
  - E.g.: AFECA, GAF, CEC, Span etc
- Density Estimation
  - Find a subset of nodes that provides a certain density threshold.
  - E.g.: ASCENT, PEAS.
- Hybrid
  - E.g.: STEM.

ASCENT

Alberto Cerpa and Deborah Estrin
UCLA

http://lecs.cs.ucla.edu/Publications/papers/ASCENT-Infocom-2002.ps

ASCENT

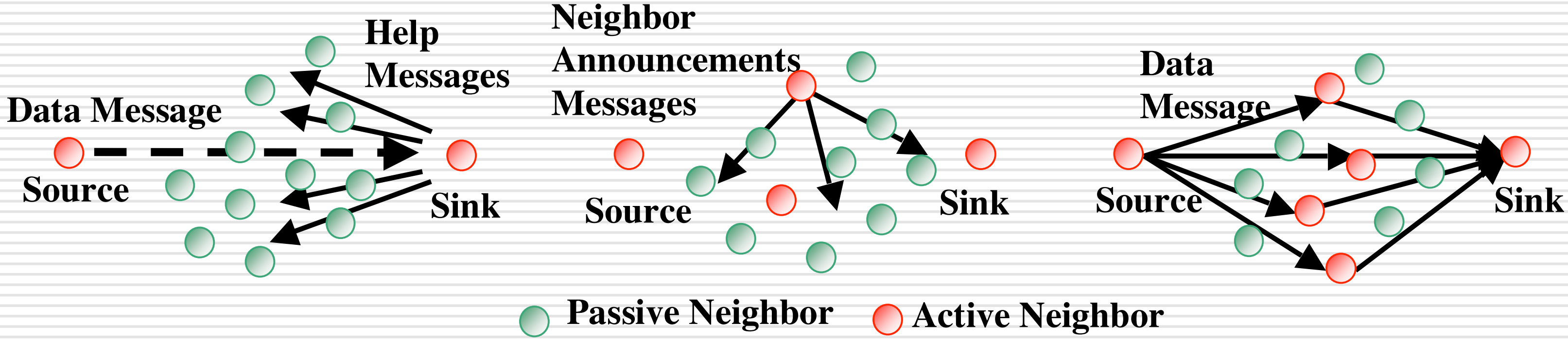
- Adaptive Self-Configuring sEnsor Networks Topologies.
- ***Observation***: different applications may require the underlying topology to have different characteristics. For example:
  - Minimal.
  - Homogeneous with a certain degree of connectivity.
  - Heterogeneous with different degrees of connectivity in different regions. Examples of these different regions may be:
    - Along a data flow path.
    - Avoiding a data flow path.
    - In the border of an event of interest.
- The goal is to exploit the redundancy in the system (high density) to save energy while providing a topology that adapts to the application needs.

Practical connectivity issues

- Wireless connectivity is a very ***complicated*** matter in the real world. Multipath effects, asymmetries, obstacles, etc. make very difficult to have a ***precise*** propagation model.
- Instead, ASCENT opts to do ***empirical adaptation***. Each node assesses its connectivity and adapts its participation into the multi-hop topology based on the measured operating region.
- Minimalist approach: ASCENT only needs to ***turn off*** the radio (sleep state) and to be able to turn the NIC/MAC in ***promiscuous mode*** (passive state).
- ASCENT runs on top of the MAC and below routing. It is independent of the routing protocol running on top, and it does not uses any information gathered by routing.

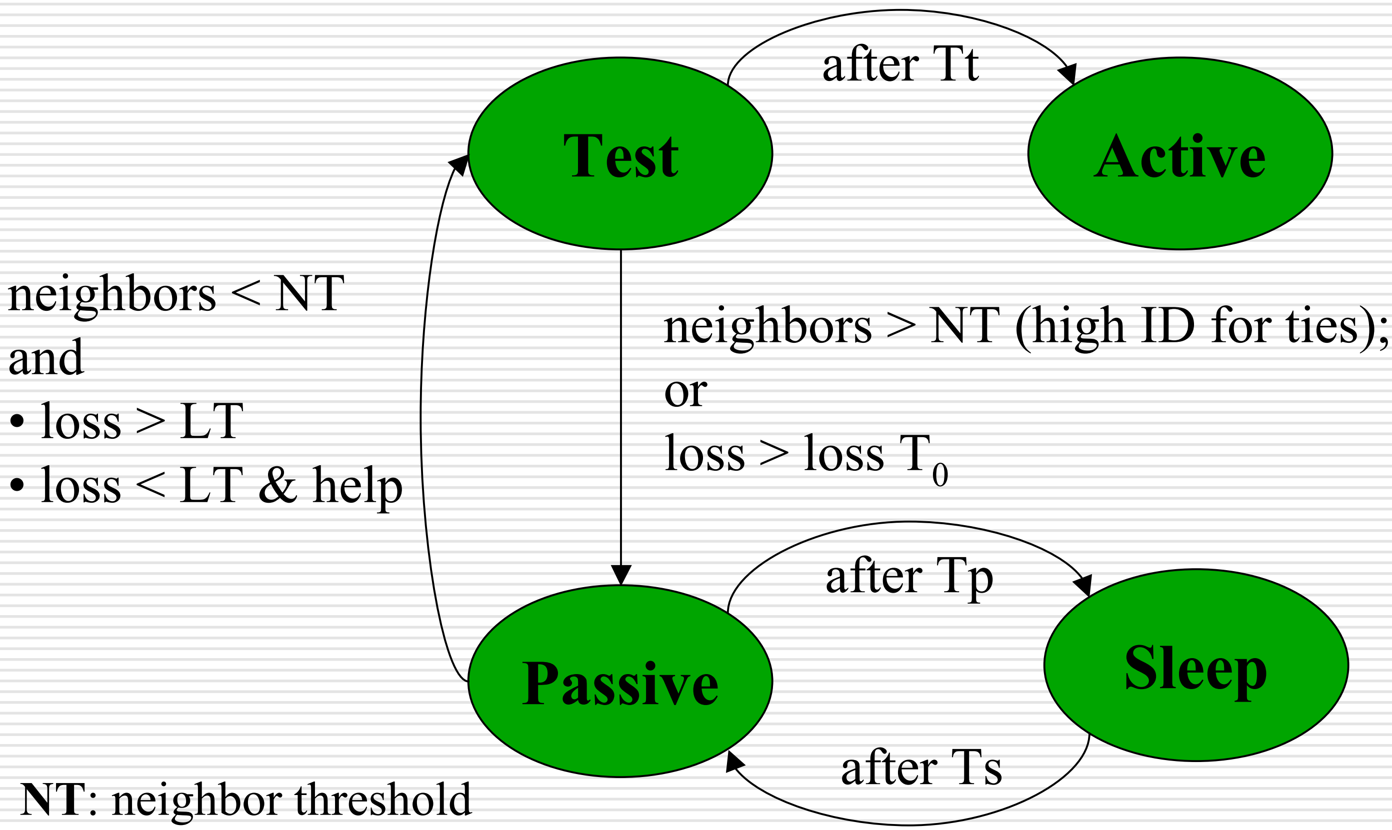
ASCENT Basics

- The nodes can be in ***active*** or ***passive*** state.
  - Active nodes are part of the topology and forward data packets (using an orthogonal routing mechanism that runs on the topology).
  - Nodes in passive state can be sleeping or collecting network measurements. They ***do not*** forward any packets.
- Each node measures the number of neighbors and packet loss ***locally***.
- Each node then makes an informed decision to ***join*** the network topology or to perform some form of ***adaptation*** (e.g. reducing its duty cycle to save energy).



(a) Communication Hole **(b) Self-configuration transition** **(c) Final State**

State Transitions



NT: neighbor threshold

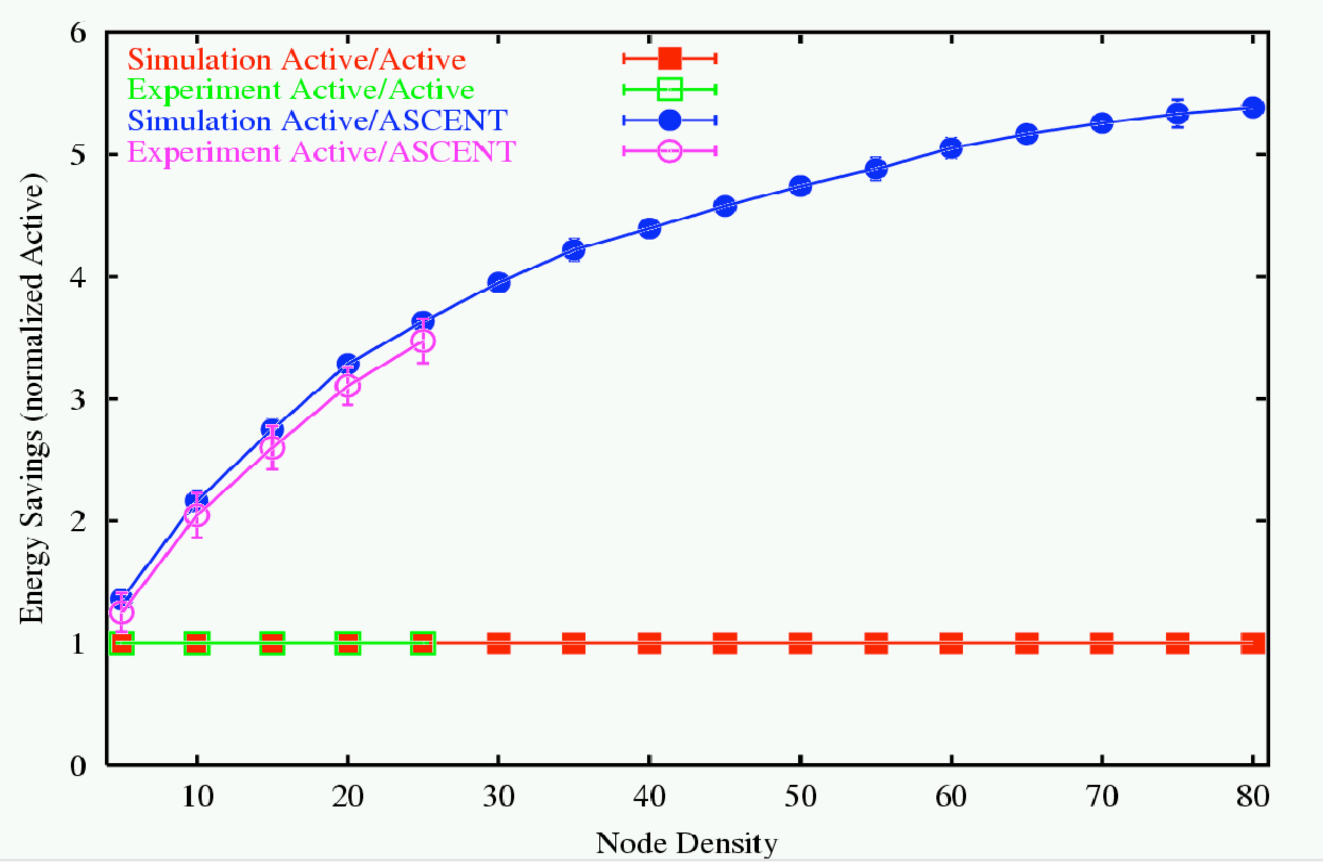
LT: loss threshold

Tx: state timer values (x = p: passive, s: sleep, t: test)

Protocol Details

- Each node adds a sequence number to each packet (this allows packet loss detection)
- Neighbor estimator: based on a neighbor loss threshold (NLT) = 1 – 1/N (N: number of neighbors in the previous cycle).
- The neighbor threshold value (NT) determines the average degree of connectivity in the network.
- The loss threshold determines the maximum amount of data loss an application can tolerate.
- Relation between Tp/Ts (passive & sleep timers) determines the amount of energy savings and convergence time in case of dynamics.

Performance Results



Energy Savings (normalized to the Active case, all nodes turn on) as a function of density. ASCENT ***provides*** significant amount of ***energy savings***, up to a ***factor of 5.5*** for high density scenarios.

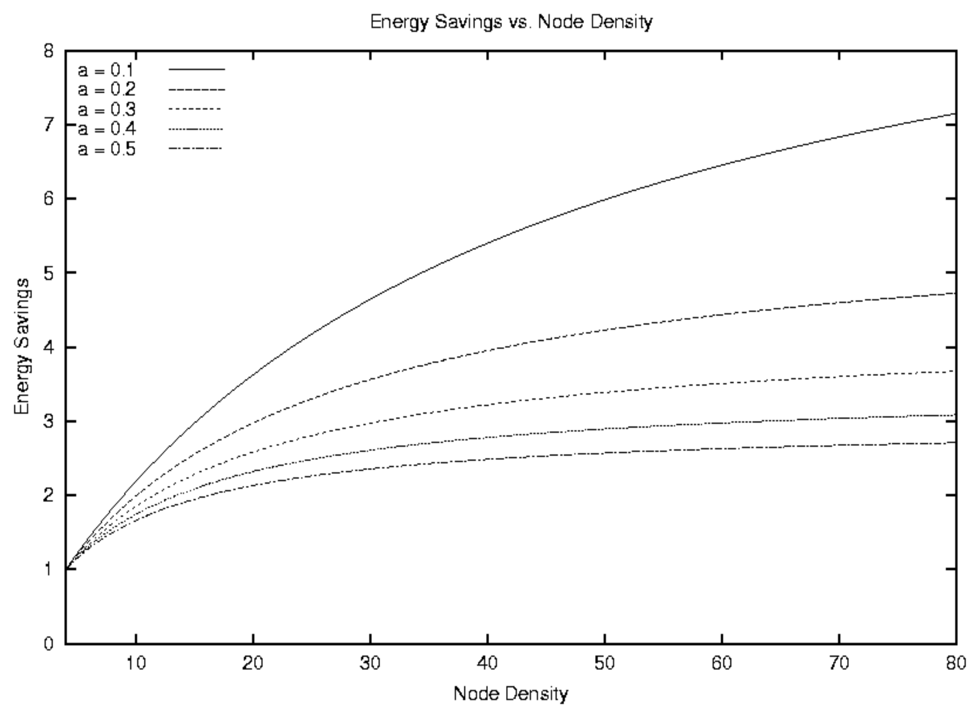
ASCENT Energy Savings Analysis

$$ES(n) = \frac{n * Idle}{NT * Idle + (n - NT) * Idle * \frac{Tp}{Tp + Ts} + (n - NT) * Sleep * \frac{Ts}{Tp + Ts}}$$

NT: neighbor threshold
Tp: passive state timer
Ts: sleep state timer
Sleep: power radio off
Idle: power radio on
$\alpha = Tp/Ts$
$\beta = Sleep/Idle = 0.004$

$$ES(n) = \frac{n}{NT + (n - NT) * \frac{\alpha + \beta}{\alpha + 1}}$$

$$\lim_{n \to \infty} ES = \frac{\alpha + 1}{\alpha + \beta}$$



Adaptive Timers

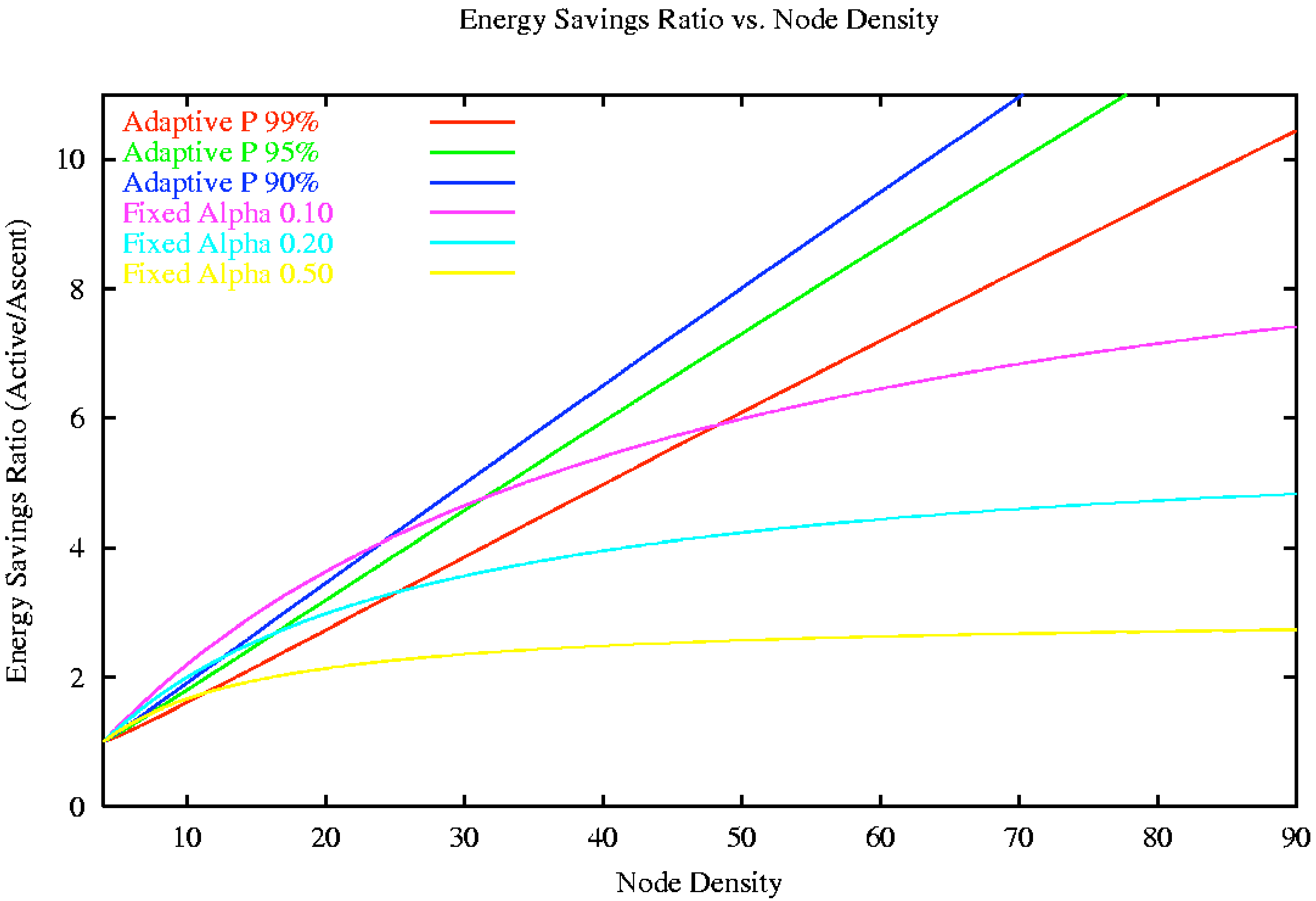
$$Pk(\text{at least k passive nodes}) = 1 - \left(\left(\frac{1}{\alpha - 1} \right)^n * \left(\frac{\alpha^k - 1}{\alpha - 1} \right) \right)$$

e.g., for k = 1;

$$\alpha(n) = 10^{-\left(\frac{\log(1 - Pk)}{n} \right)} - 1$$

- For any given probability target Pk and given the number of passive nodes in the area, nodes could calculate the optimal relation between the passive and sleep timers (α)
- The larger the Pk target, the larger the alpha for any given density.
- The larger the k, the larger the alpha (although it grows VERY slowly).

Adaptive vs Fixed Timers



Ascent: Pros and Cons

- Pros:
  - Empirical adaptation is robust to RF uncertainties
- Cons:
  - Lots of protocol overhead (true for most adaptive topology schemes)
  - Lots of parameter settings whose impact on system behavior is not clear
    - Neighbor threshold, loss threshold, neighbor loss threshold, etc

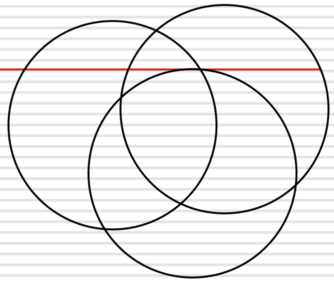
Comparison of Adaptive Topology Schemes

	Goal (energy savings)	Routing dependency	Assumptions
GAF/CEC	preserve routing fidelity	none	geographic information for grid placement radio connectivity directly correlated with geography (gaf)
SPAN	preserve capacity of the raw topology	gets connectivity matrix and neighbors from routing requires modifications in the routing lookup process	802.11 MAC with Power Savings mode
STEM	tradeoff latency for energy savings	needs routing info to direct the wake-up wave	2 radios/wake-up channel connectivity conditions remain constant in sleeping periods
ASCENT	adapt topology based on application needs	none	radio supports promiscuous mode

Discussion: MAC-layer scheduling vs Adaptive Topology Management

- Both schedule wakeup and listen times to lower power consumption. Why do we need both?
- Compare schemes in terms of:
  - Achievable power gains
  - Scheduling complexity
- If both schemes were employed simultaneously, how would they interact?

Sensor Placement Constraints



- Sensing constraints
  - Discrete set of points:
    - Percentage of desired (known a-priori) measurement points covered
    - Percentage of the operational area that is covered within a sensing range S of k-sensors
  - Continuous monitoring of sensor field:
    - Average/Max sensing distortion bound
- Network coverage constraints:
  - Density of placement (average/max distance between nodes)
- Area and terrain characteristics
  - Terrain map of set of points/area where nodes can actually be placed.

Maximal Breach and Maximal Support

S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M.B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks", *INFOCOM* 2001.

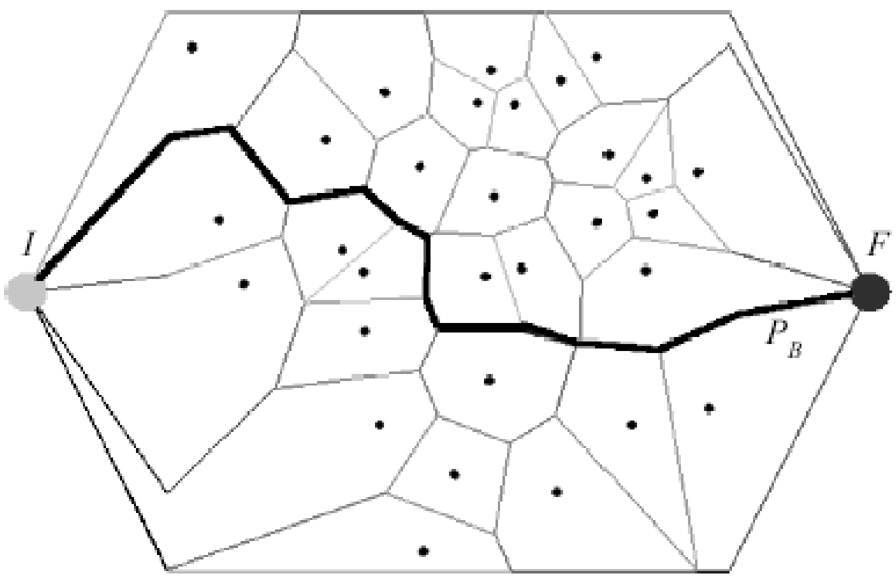


Figure 4 - Sensor Field With Weighted Voronoi Diagram And Maximal Breach Path

Maximal breach path: path of an object which maximizes the closest approach to any sensor

Calculated using graph algorithms based on Voronoi diagrams

Maximal Breach and Maximal Support

S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M.B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks", *INFOCOM* 2001.

Maximal support path: path of an object which minimizes the farthest distance from any sensor

Calculated using graph algorithms based on Delaunay triangulation

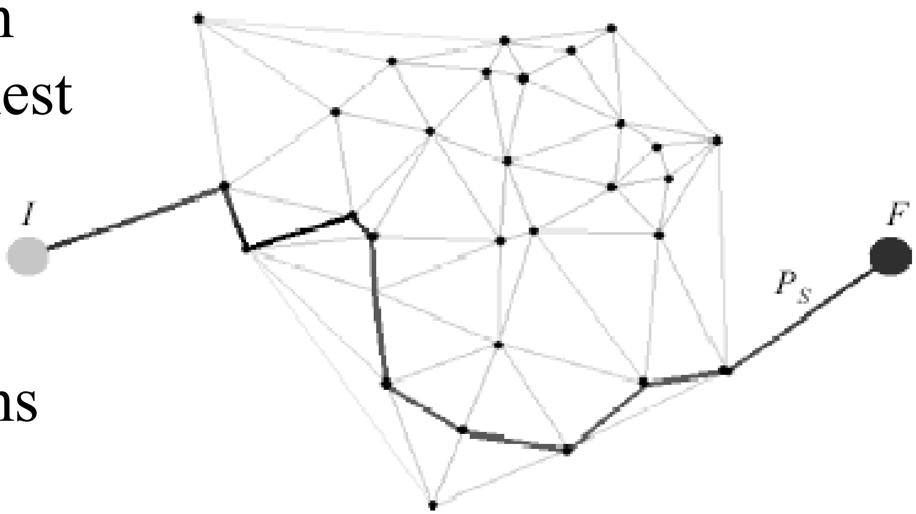
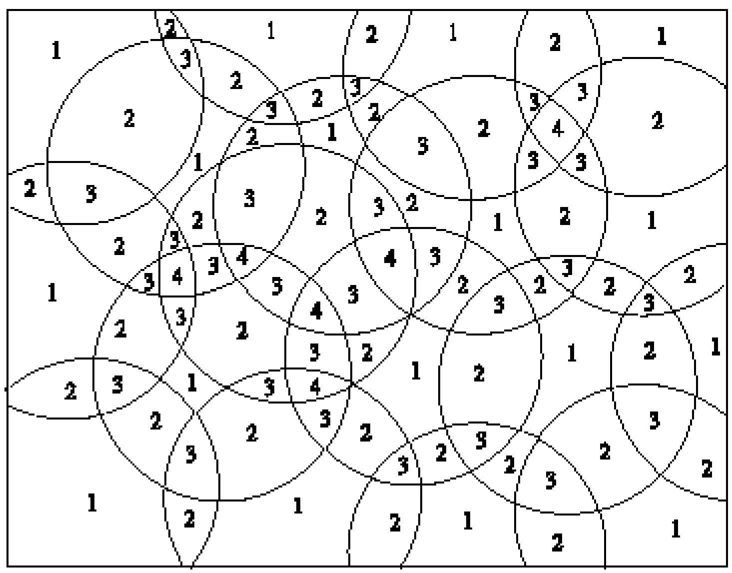


Figure 5 - Sensor Field With Weighted Delaunay Triangulation And Maximal Support Path (P_S)

K-Coverage

C. Huang and Y. Tseng. The coverage Problem in a Wireless Sensor Network. *WSNA03*, San Diego, CA. 2003.

Theorem 1: The whole operational area is k-covered, if and only if each sensor in the area is k-perimeter-covered



Coverage and Connectivity

- X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks. Sensys03, Los Angeles, CA, USA. November, 2003.

Theorem 2: The whole operational area is k-covered, if and only if every (non-empty) intersection between sensors and between sensors and the boundary is k-covered.

Theorem 3: A set of nodes that k-cover a convex region A forms a k-connected network graph if Rc >= 2Rs

(for interior nodes, this actually provides 2k-connectivity)