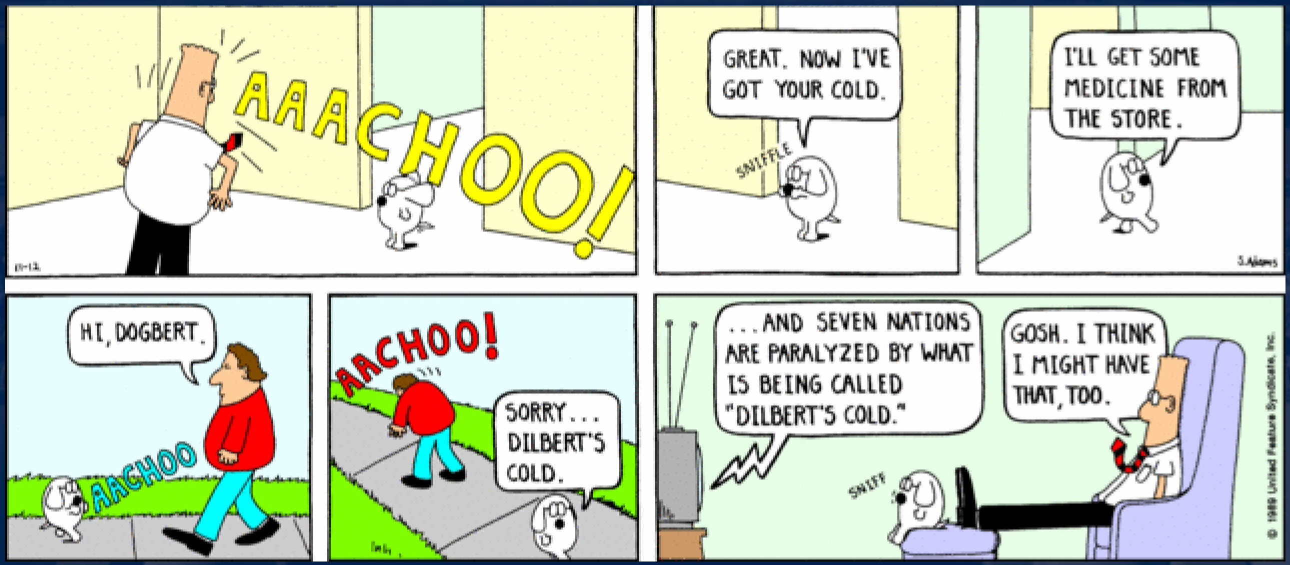
UNIVERSITY OF ILLINOIS

AT URBANA-CHAMPAIGN

# Spreading the Rumor




Mainak Ghosh and Mayur Sadavarte

1867

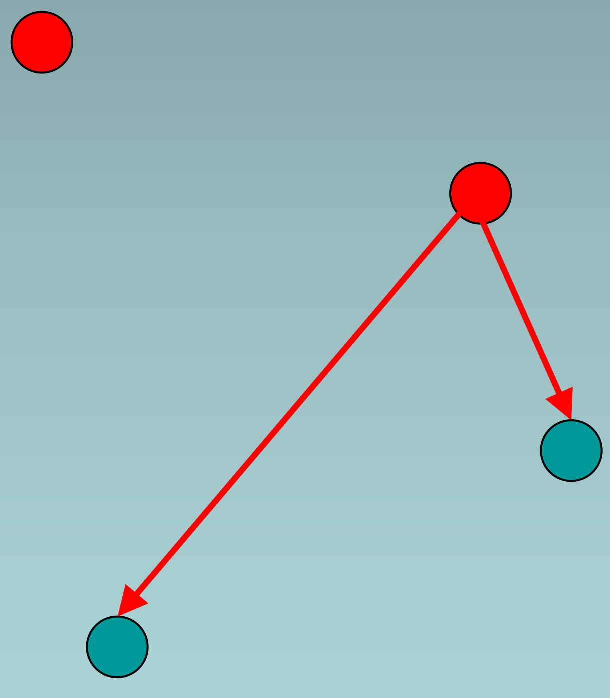
illinois.edu

Courtesy: http://dilbert.com/strips/comic/1989-11-12/

# Sounds Familiar

Infective

**Gossip based Multicast**



Susceptible

Slide Borrowed from Indy’s Introduction Presentation

# It’s Not Rumored

- Clearinghouse and Bayou projects: email and database transactions [PODC ‘87]
- refDBMS system [Usenix ‘94]
- Bimodal Multicast [ACM TOCS ‘99]
- Sensor networks [Li Li et al, Infocom ’02, and PBBF, ICDCS ‘05]
- Usenet NNTP (Network News Transport Protocol) ! [ ‘79]

Slide Borrowed from Indy’s Introduction Presentation

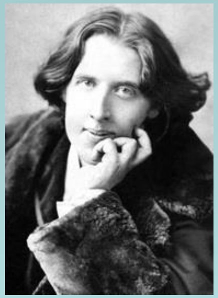
# EPIDEMIC ALGORITHMS FOR REPLICATED DATABASE MAINTENANCE

Alan Demers et al, PODC 1987
Presenter: Mainak Ghosh

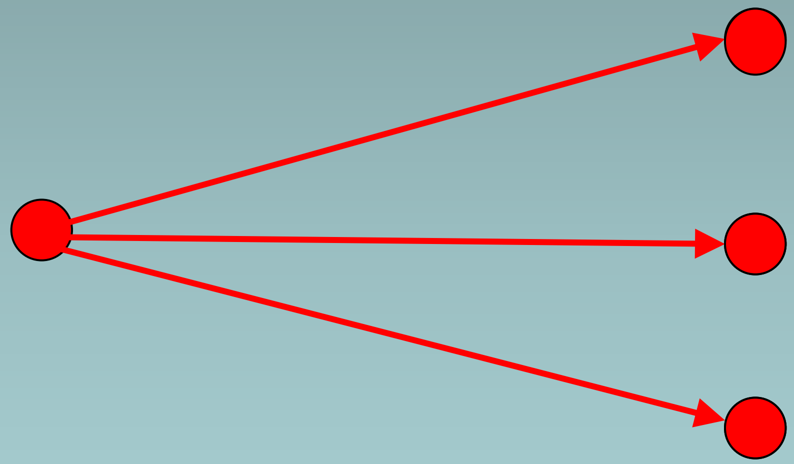
# Consistency

- Replicated Data = Consistency Issues
- System Model: Underlying communication system unreliable
- Goal: Replicas should be eventually consistent.
- Solution: Randomized Algorithms inspired from Epidemics

"Consistency is the hallmark of the unimaginative."
- Oscar Wilde



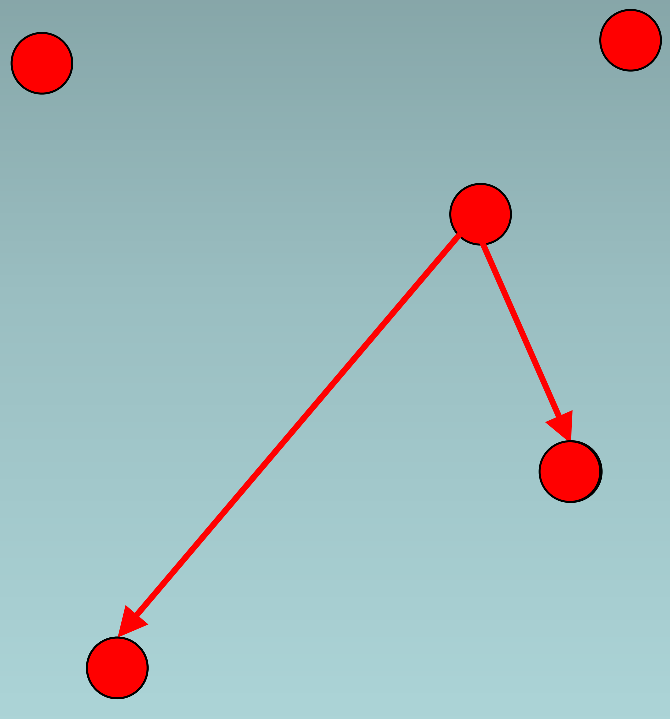
# Direct Mail



Cons??

Overhead??

# Anti Entropy (Push)

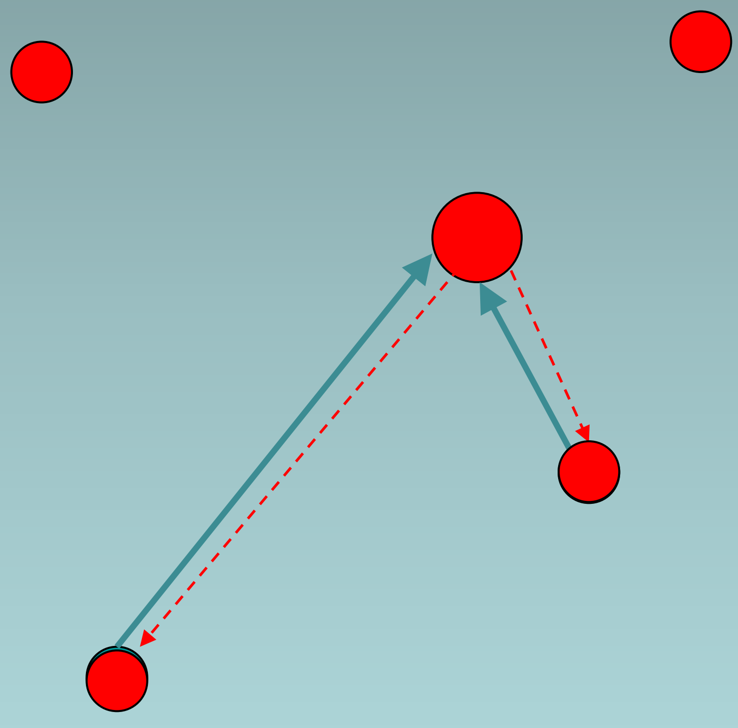


$p_i$- Probability that a node is susceptible after $i_{th}$ round
n – number of sites

$$p_{i+1} = p_i(1 - \frac{1}{n})^{n(1 - p_i)}$$

Converges slowly to zero for small $p_i$ and large n

# Anti Entropy (Pull)



$p_i$- Probability that a node is susceptible after $i_{th}$ round

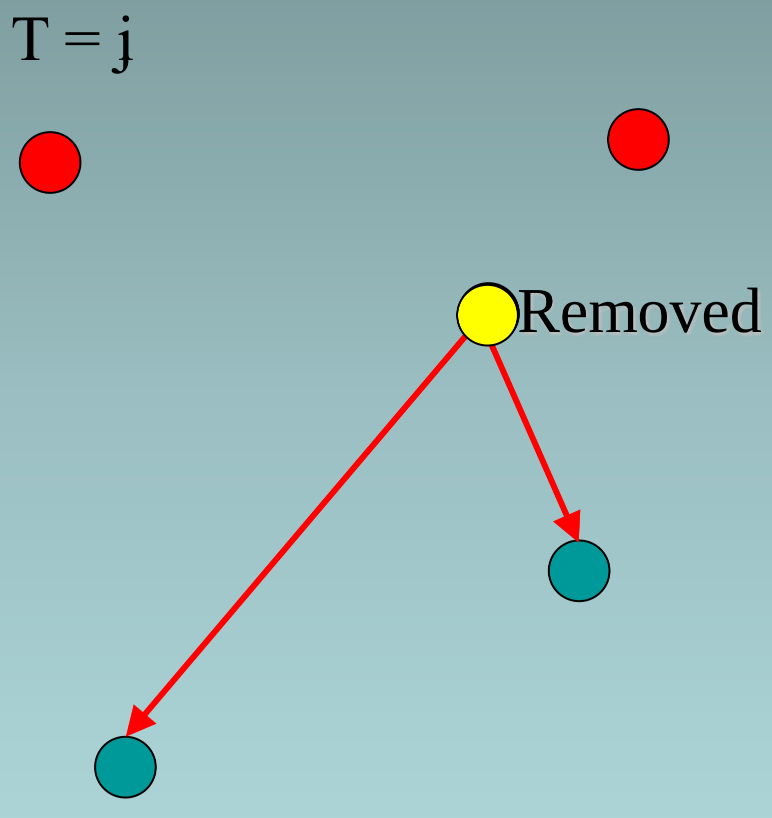
$$p_{i+1} = (p_i)^2$$

Converges rapidly to zero for small $p_i$

# Anti Entropy (Optimizations)

- Maintain checksum, compare databases if checksums unequal
- Maintain recent update lists for time *T*, exchange lists first
- Maintain inverted index of database by timestamp; exchange information in reverse timestamp order, incrementally re-compute checksums

# Stale Gossip




- List of infective updates maintained at sites
- Complexity involved in choosing when to remove from the list

# Epidemic Variants

- Blind vs. Feedback
  - Blind: lose interest to gossip with probability *1/k* every time you gossip
  - Feedback: Loss of interest with probability *1/k* only when recipient already knows the rumor
- Counter vs. Coin
  - Coin: above variants
  - Counter: Lose interest completely after *k* unnecessary contacts. Can be combined with blind.
- Push vs. Pull

# Performance Metrics

- Residue: Fraction of susceptible left when epidemic finishes
- Traffic: (Total update traffic) / (No. of sites)
- Delay: Average time for receiving update and maximum time for receiving update

# Performance Evaluation

**Table 1.** Performance of an epidemic on 1000 sites using response and counters.

| Counter $k$ | Residue $s$ | Traffic $m$ | Convergence $t_{ave}$ | $t_{last}$ |
|---|---|---|---|---|
| 1 | 0.176 | 1.74 | 11.0 | 16.8 |
| 2 | 0.037 | 3.30 | 12.1 | 16.9 |
| 3 | 0.011 | 4.53 | 12.5 | 17.4 |
| 4 | 0.0036 | 5.64 | 12.7 | 17.5 |
| 5 | 0.0012 | 6.68 | 12.8 | 17.7 |

**Table 2.** Performance of an epidemic on 1000 sites using blind and probabilistic.

| Counter $k$ | Residue $s$ | Traffic $m$ | Convergence $t_{ave}$ | $t_{last}$ |
|---|---|---|---|---|
| 1 | 0.960 | 0.04 | 19 | 38 |
| 2 | 0.205 | 1.59 | 17 | 33 |
| 3 | 0.060 | 2.82 | 15 | 32 |
| 4 | 0.021 | 3.91 | 14.1 | 32 |
| 5 | 0.008 | 4.95 | 13.8 | 32 |

# Death Certificate

- Deleted items may get resurrected!
- Use of death certificates (DCs) – when a node receives a DC, old copy of data is deleted
- How long to maintain a DC?
  - Typically twice (or some multiple of) the time to spread the information
  - Alternately, use Chandy and Lamport snapshot algorithm to ensure all nodes have received
  - Certain sites maintain *dormant* DCs for a longer duration; re-awakened if item seen again

# Spatial Distributions

- Cost of communication is not uniform across all sites
- Sites choose nearby neighbors to run their protocol
- Results:
  - Critical Links get less traffic
  - Protocol converge with little change in total generated traffic

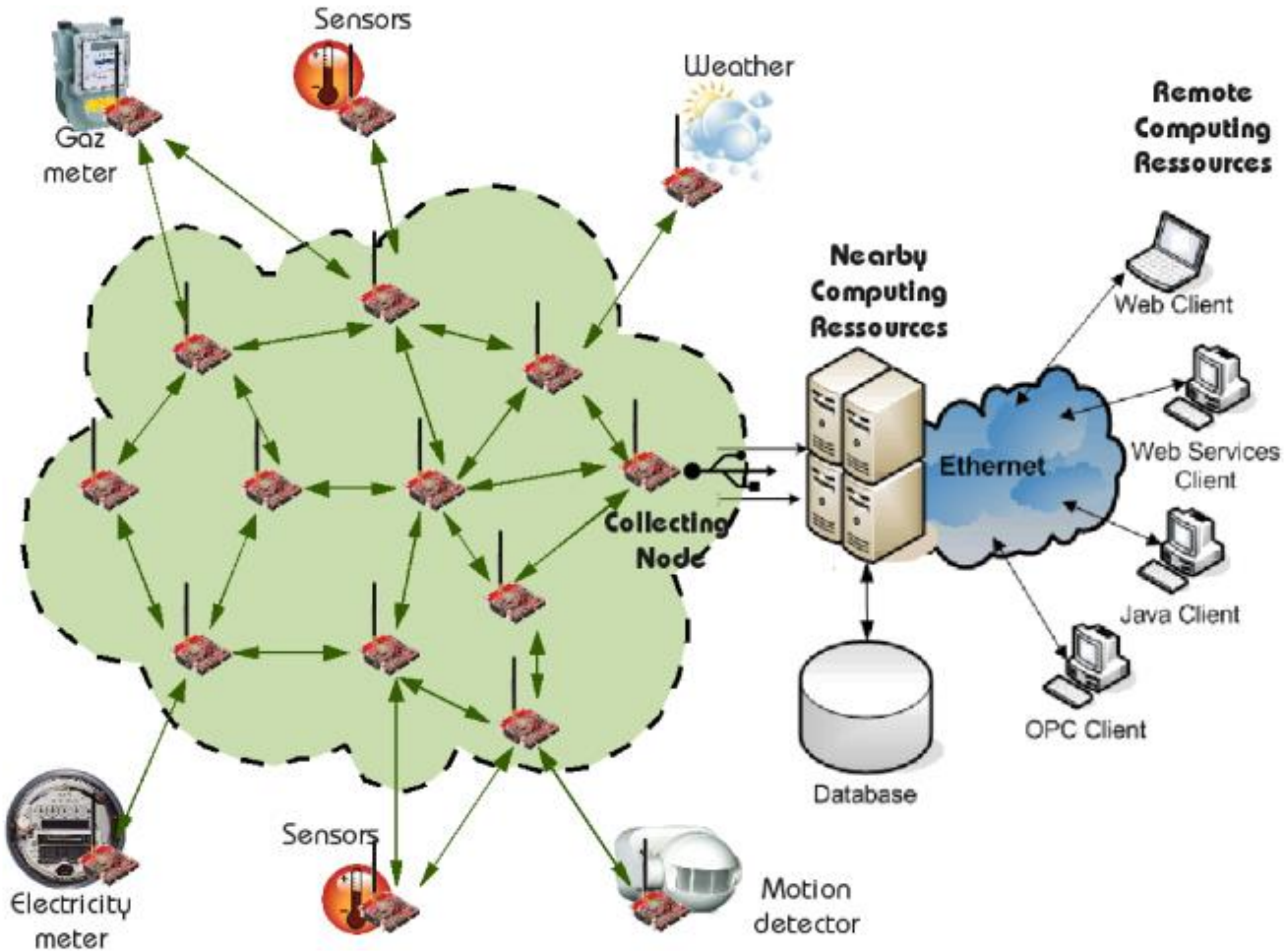
# Discussion

- Anti Entropy Optimization Strategies
- Death Certificate Removal
- Gossips in OSNs, other areas…

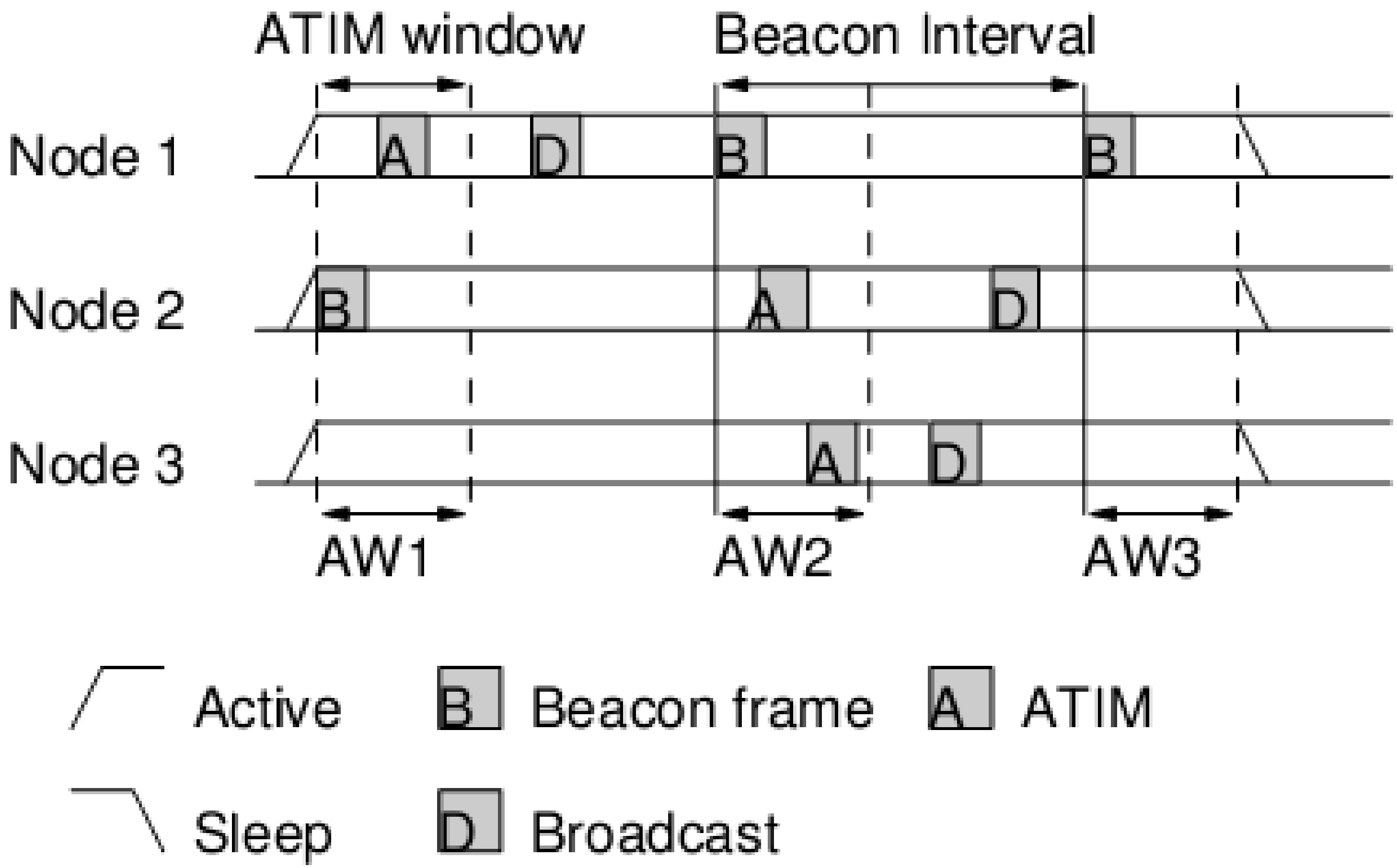
# Exploring the Energy-Latency Trade-Off for Broadcasts in Energy-Saving Sensor Networks

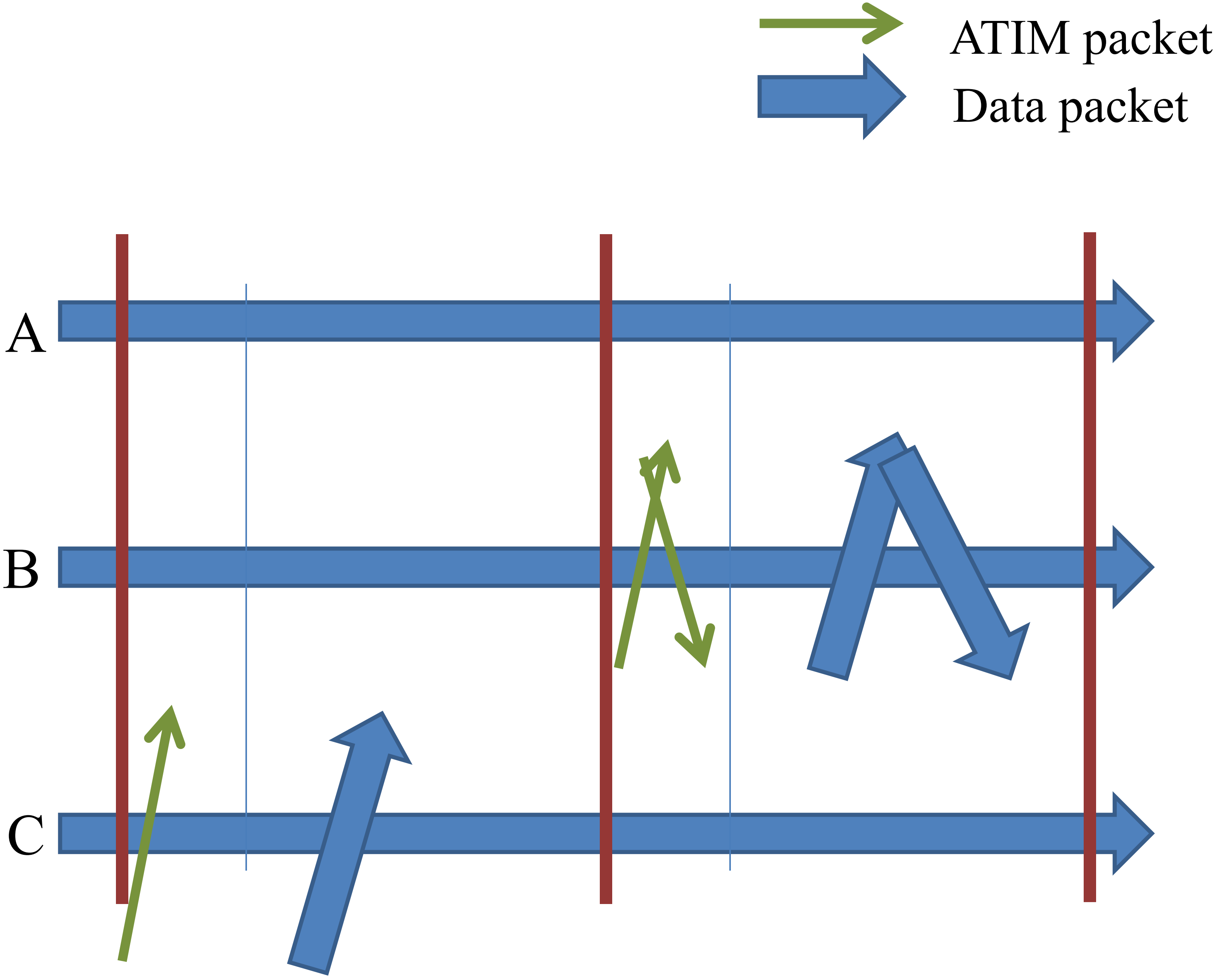
- Author(s): Miller, Matthew J. ; Sengul, Cigdem ; Gupta, Indranil (Dept. of Comput. Sci., Illinois Univ., Urbana-Champaign, IL)
- 25th IEEE International Conference on Distributed Computing Systems.
- Identifier: 10.1109/ICDCS.2005.35
- Publication Year: 2005
- Presenter: Mayur Sadavarte

# Sensor Networks



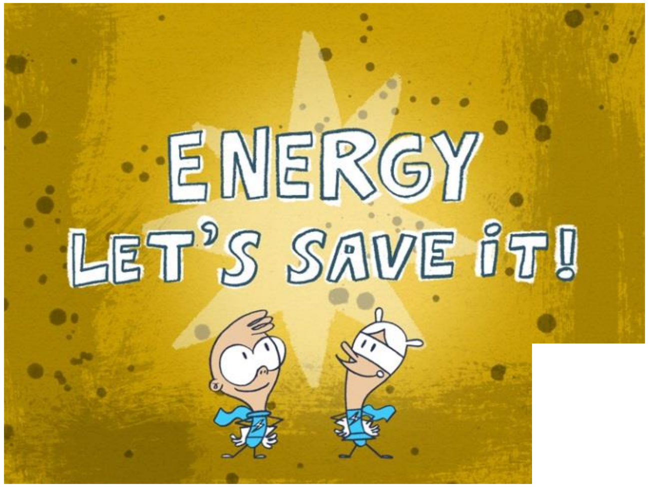
# Active-Sleep Cycle Approach

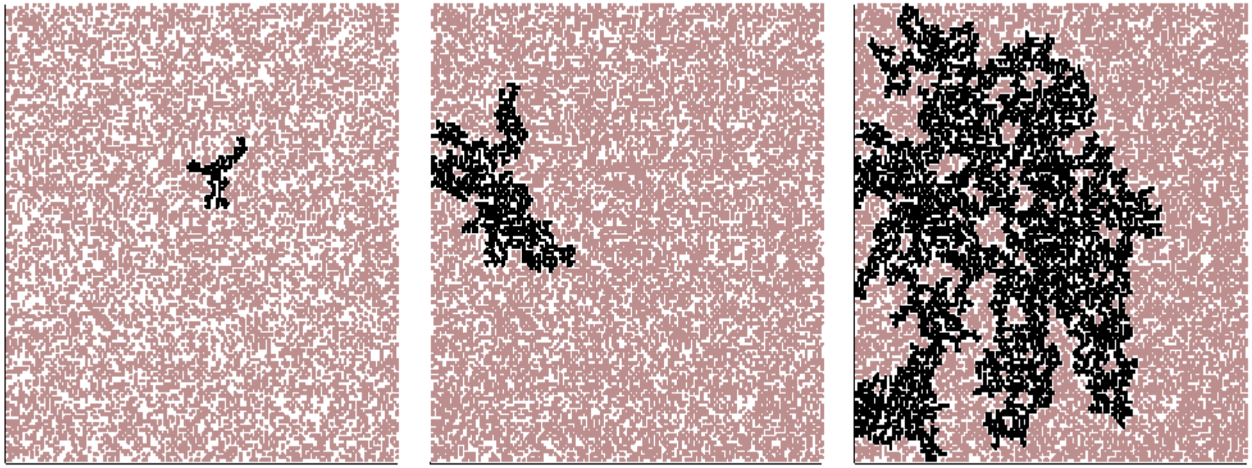



ATIM packet
Data packet
A
B
C

# Trinity to Optimize

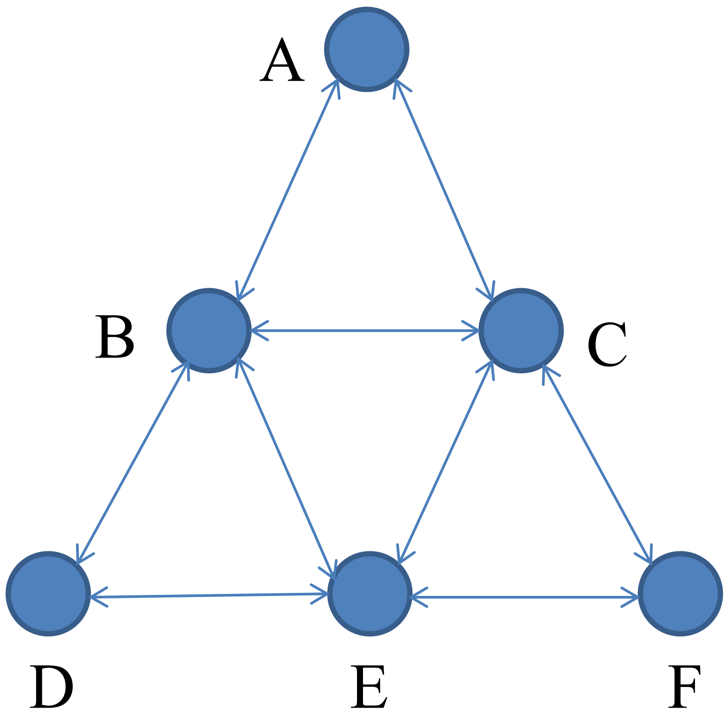
- Energy
- Latency
- Reliability



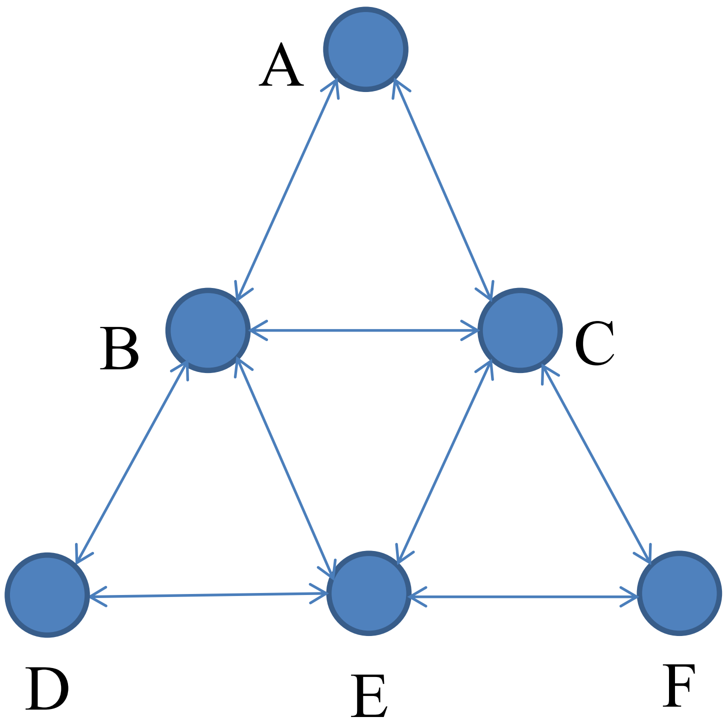



# Probability Based Approach

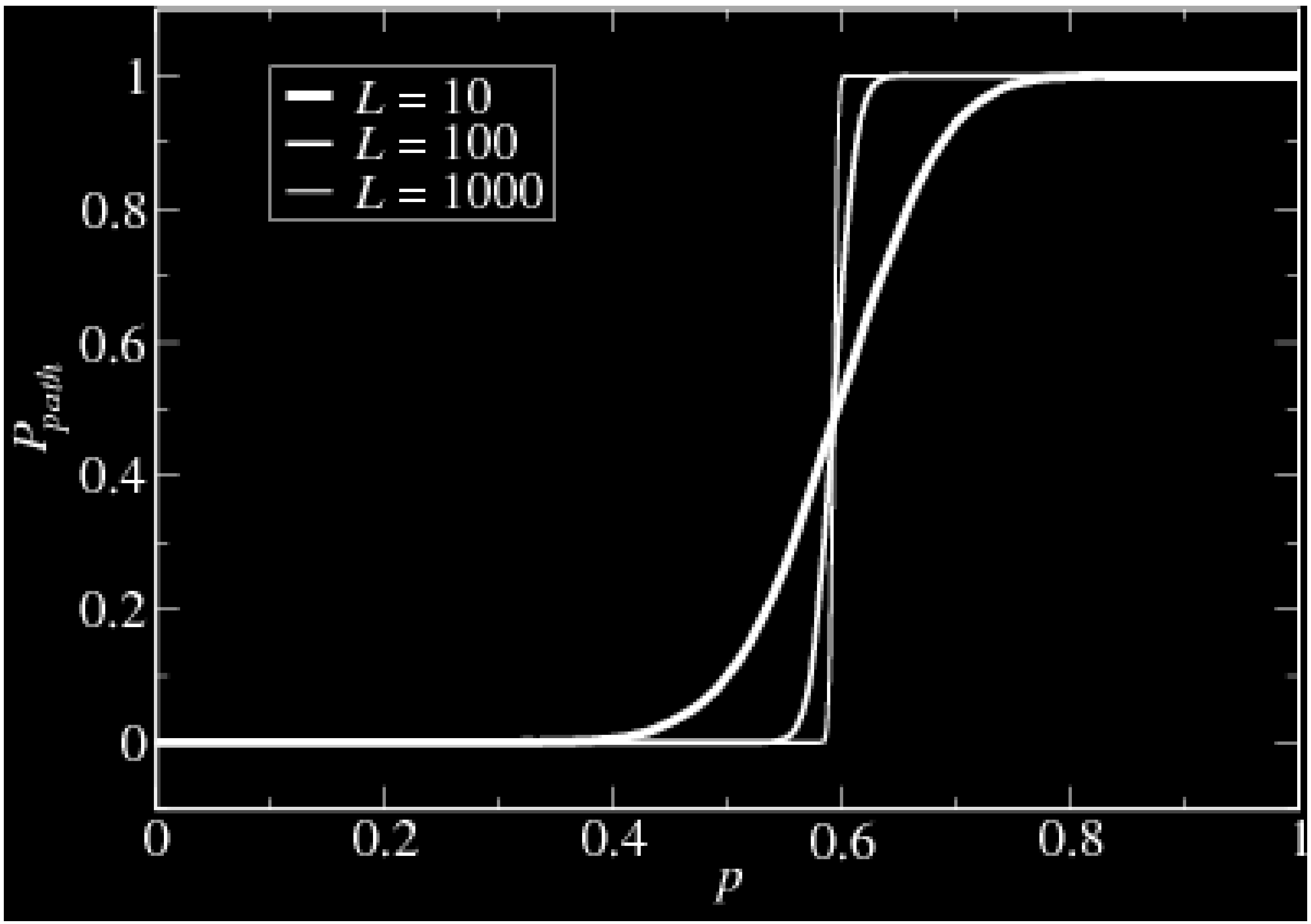
Site Percolation




Bond Percolation



# Percolation Theory Result



# PBBF

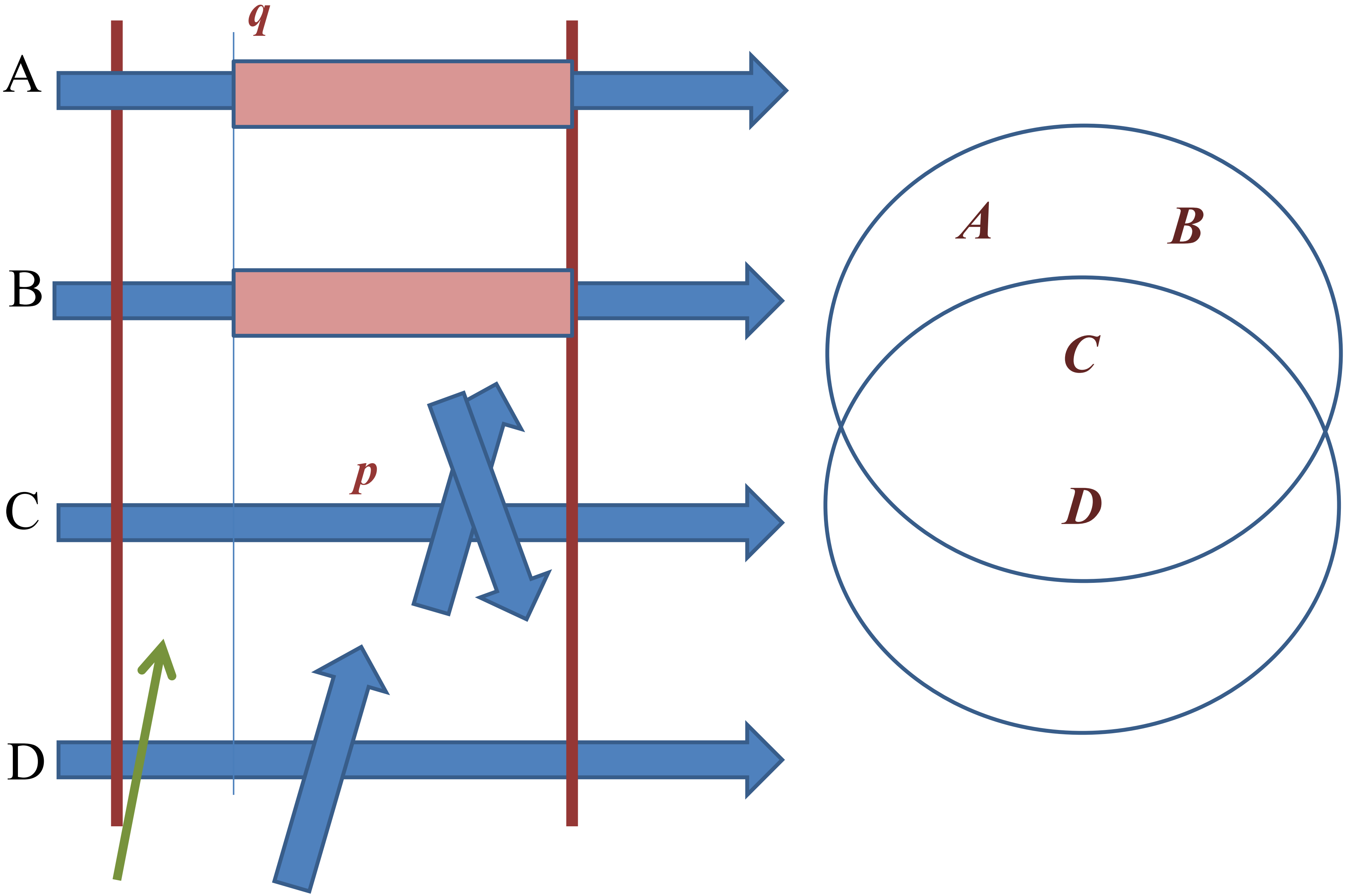
Distinguishing Points –

- Bond percolation probability model
  - Gossip-based routing protocol proposed by Z. J. Haas, J. Y. Halpern, and L. Li in Gossip-Based Ad Hoc Routing, is based on 'site-percolation model'
- Operates in close proximity with MAC layer protocol
- Range of operating points, based on energy-latency tradeoff for different levels of reliability, from which an application designer can choose.

# Trade-Off Knobs

- ***p:*** probability that node rebroadcasts a packet though not all the neighbors are guaranteed to be awake to receive the packet

- ***q:*** probability that node keeps its radio on even after the active time, when it is actually supposed to sleep.

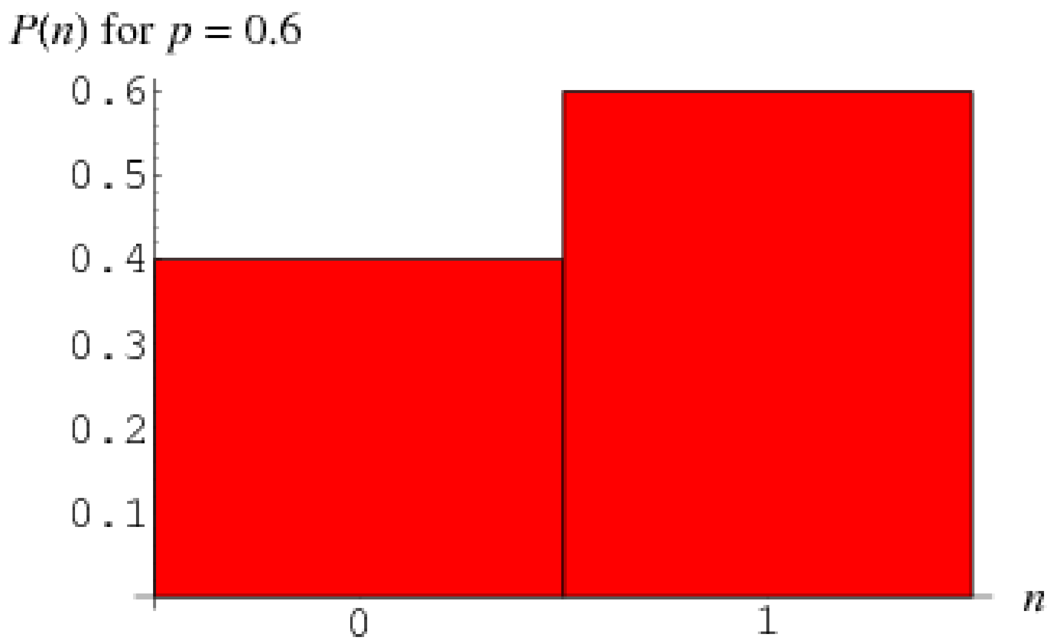

q
A
B
p
C
D
A
B
C
D

# ‘*p*’ & ‘*q*’

- *p* – presents trade-off between latency and reliability
  - As p increases, latency decreases while the fraction of nodes not receiving a broadcast increases (unless q = 1)
- *q* – presents trade-off between energy and reliability
  - As q increases, energy consumption increases, but the fraction of nodes receiving a broadcast increases (unless p = 0)

# $p_{edge}$

- 'mean' of a Bernoulli Random Variable which governs a state for individual edge in the graph G




- $p_{edge}$ - '$pq + (1-p)$'

# Critical Probability ($P_c^{bond}(G)$)

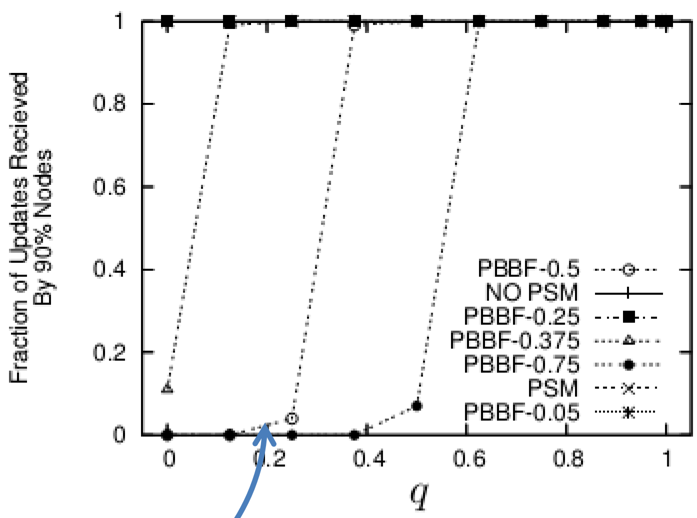
Consider *G(V, E)* to be an infinite connected graph, and $n_0$ to be source of Gossip.

$$C_0 = \{x \in V : n_0 \leftrightarrow x\}.$$

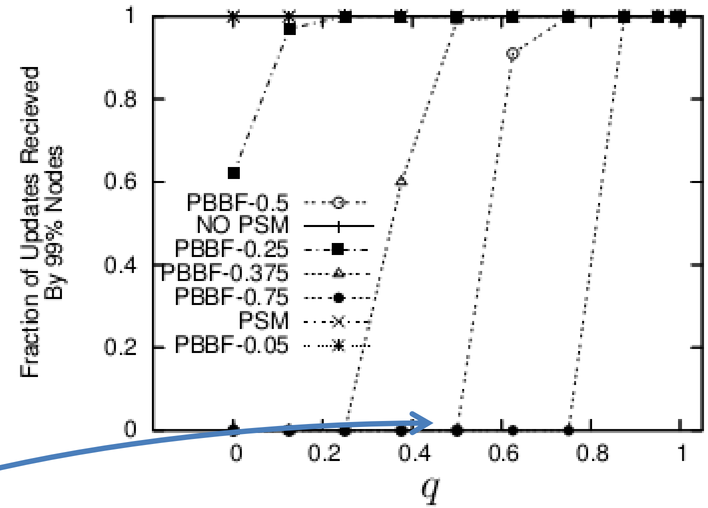
We want $C_0$ to be infinite !!

$$p_c^{bond}(G) = sup\{p_{edge} : \theta^{bond}(p_{edge}) = 0\}$$

# Reliability (most important)

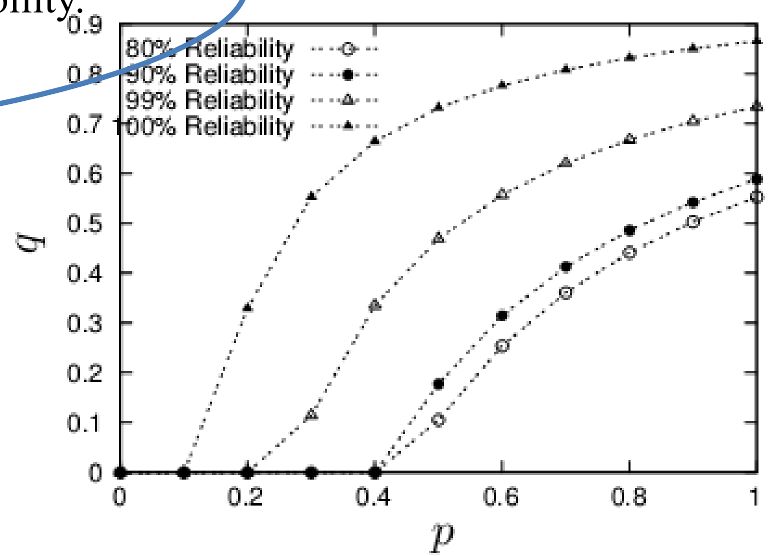



Threshold behavior for 90% reliability.




Threshold behavior for 99% reliability.

Threshold Behavior

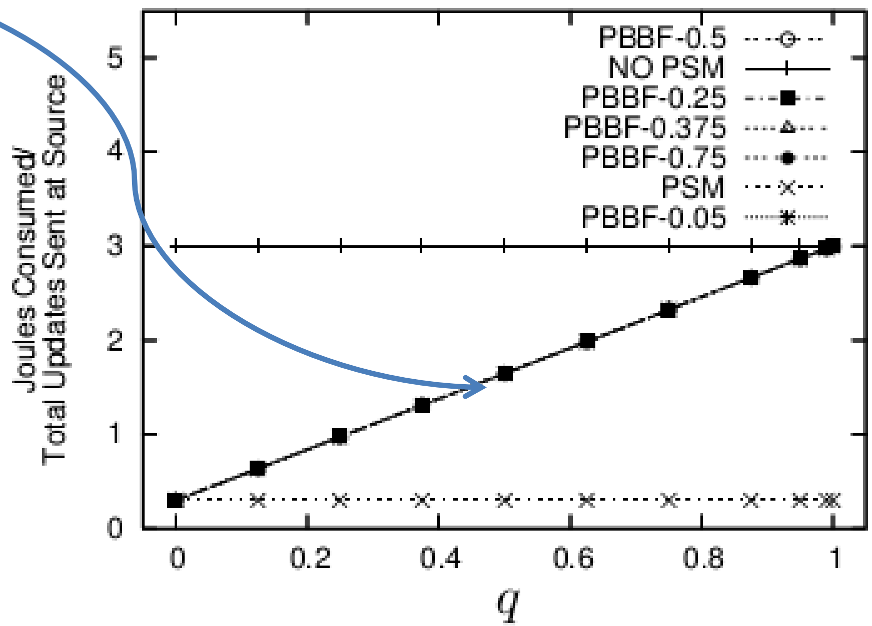



Relationship between p and q for a given reliability level in a 30 × 30 grid network.

# Energy (effect of ‘*q*’)

‘*p*’ doesn’t affect energy consumption

$$E_{PBBF} = \frac{T_{active} + q \cdot T_{sleep}}{T_{frame}}$$



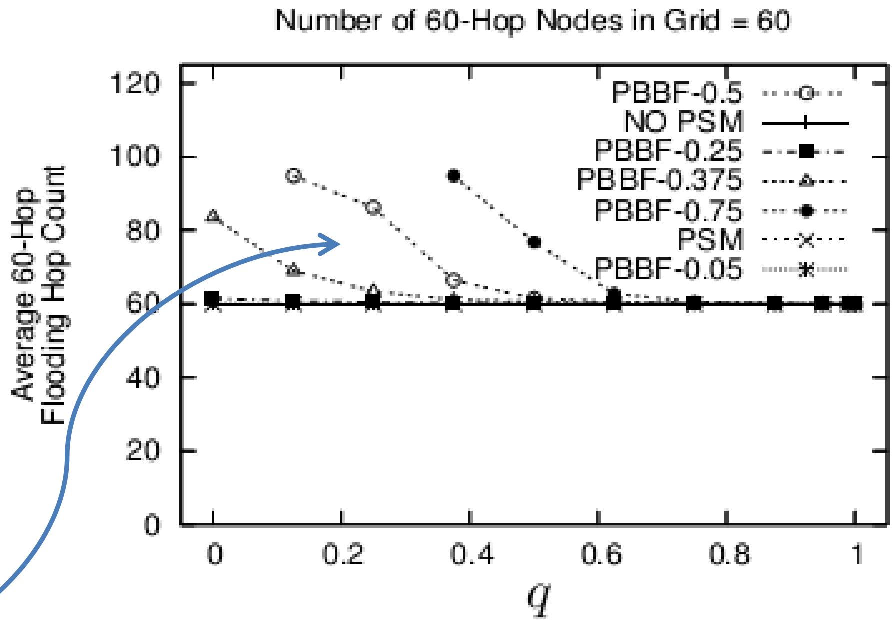

Average energy consumption.

$$\frac{E_{PBBF}}{E_{original}} = \frac{T_{active} + q \cdot T_{sleep}}{T_{active}} = 1 + q \cdot \frac{T_{sleep}}{T_{active}}$$

# Latency

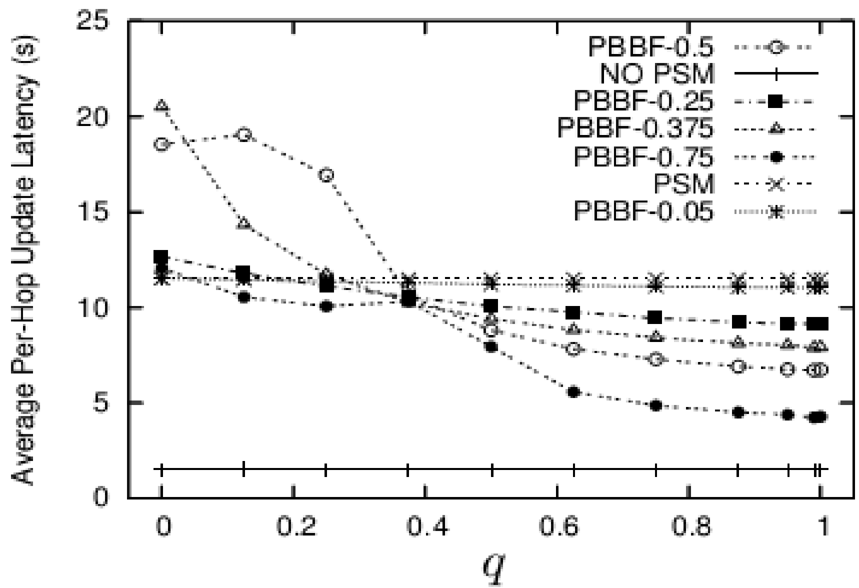
- ***L1***: time required to actually transmit and receive the packet
  - Depends upon the MAC protocol
- ***L2***: time required to wake up all neighbors for broadcast
  - Depends upon the sleep-scheduling mechanism

$$L = \frac{L_1 \cdot p \cdot q + (L_1 + L_2) \cdot (1-p)}{p \cdot q + (1-p)}$$
$$= L_1 + L_2 \cdot \frac{1-p}{1-p+p \cdot q}$$



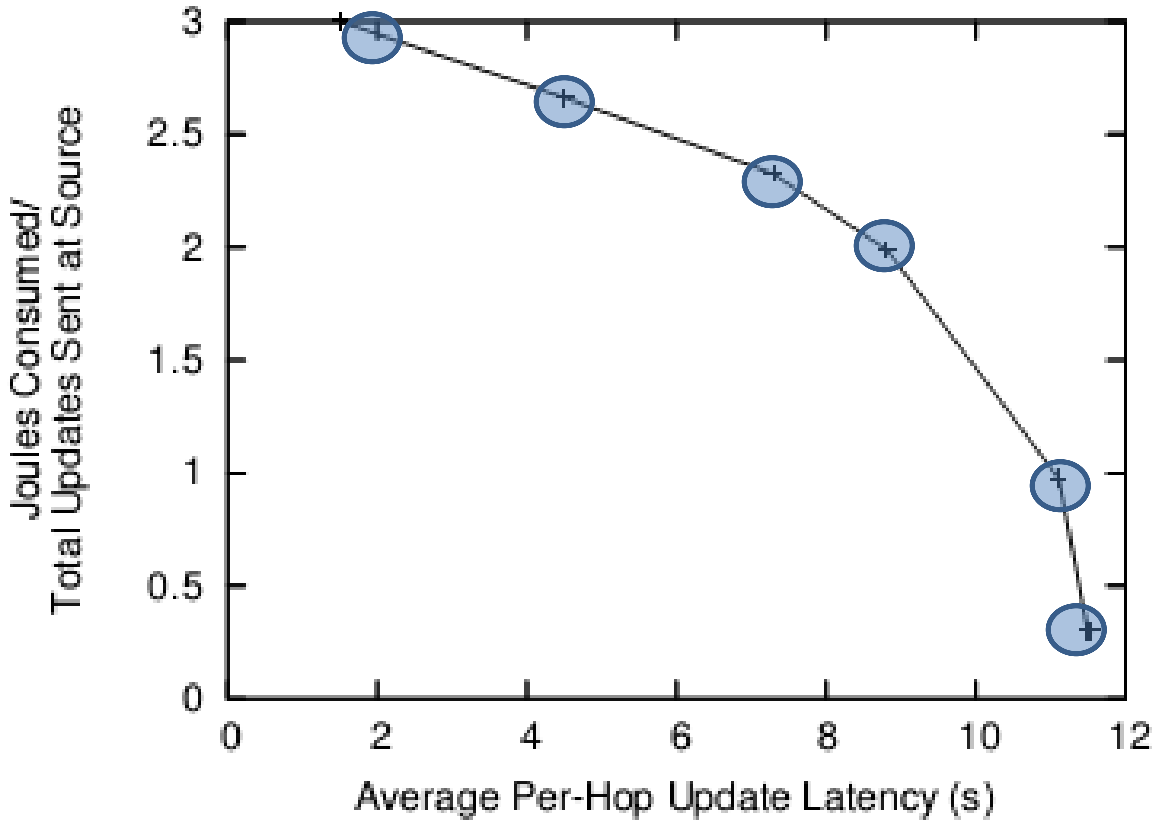

Average hops traveled by an up-date to reach a node 60 hops from the source.

Shortest Path gets hindered due to probabilistic edges




Average per-hop update latency.

# Energy – Latency Trade-off

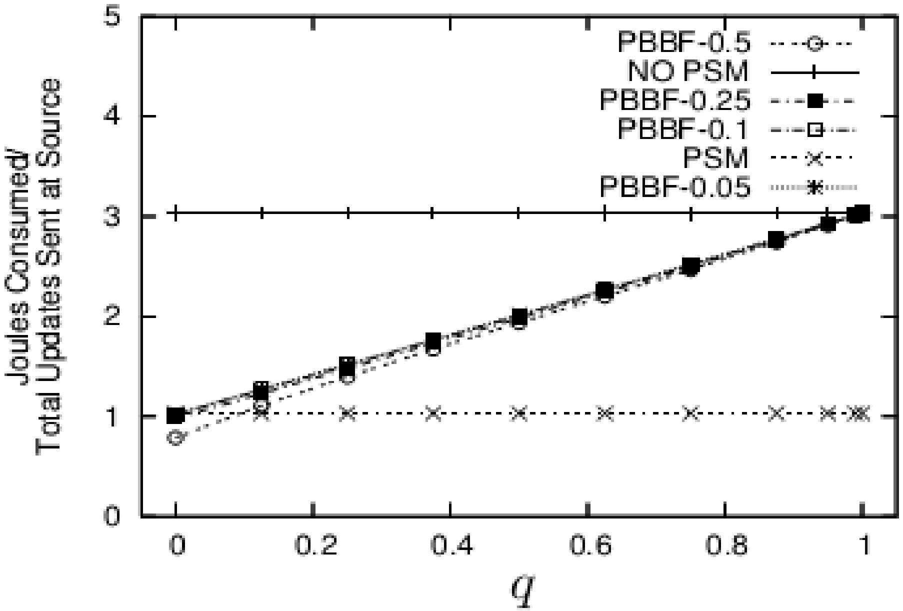



Energy-latency trade-off for 99% reliability.

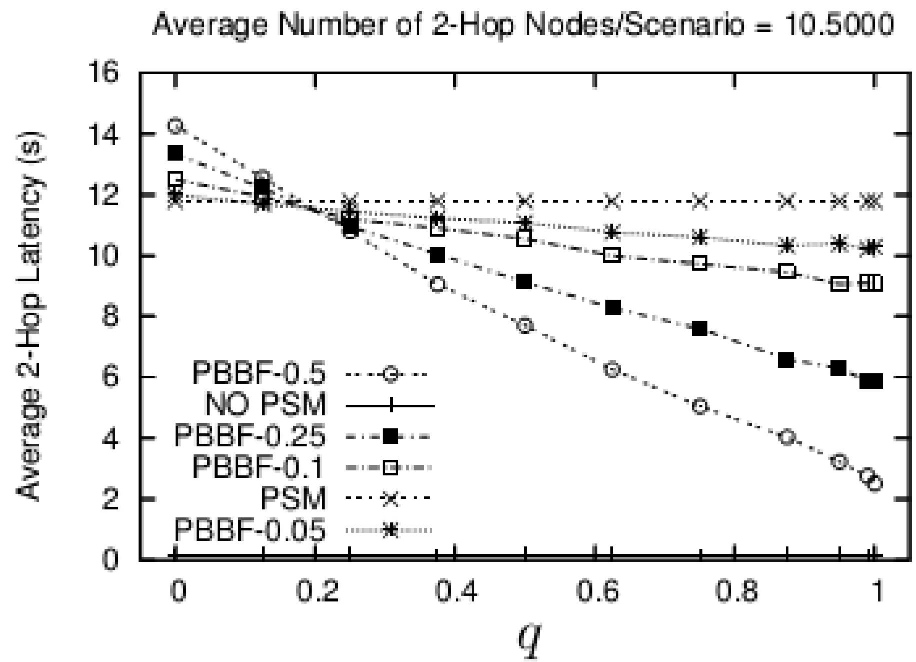
# Simulation Setup

- IEEE 802.11 PSM MAC using ns-2 simulator
- With collisions and interference
- Code distribution in sensor network application
- Perfect sync across the whole sensor n/w is assumed
- N: 50
- $\Delta$ : $(\pi R^2)*N/A$
- $\lambda$ : broadcast rate – 0.01 packets/s
- $T_{frame}$ : 10 s
- $T_{active}$ : 1 s

# Impact of '*q*'

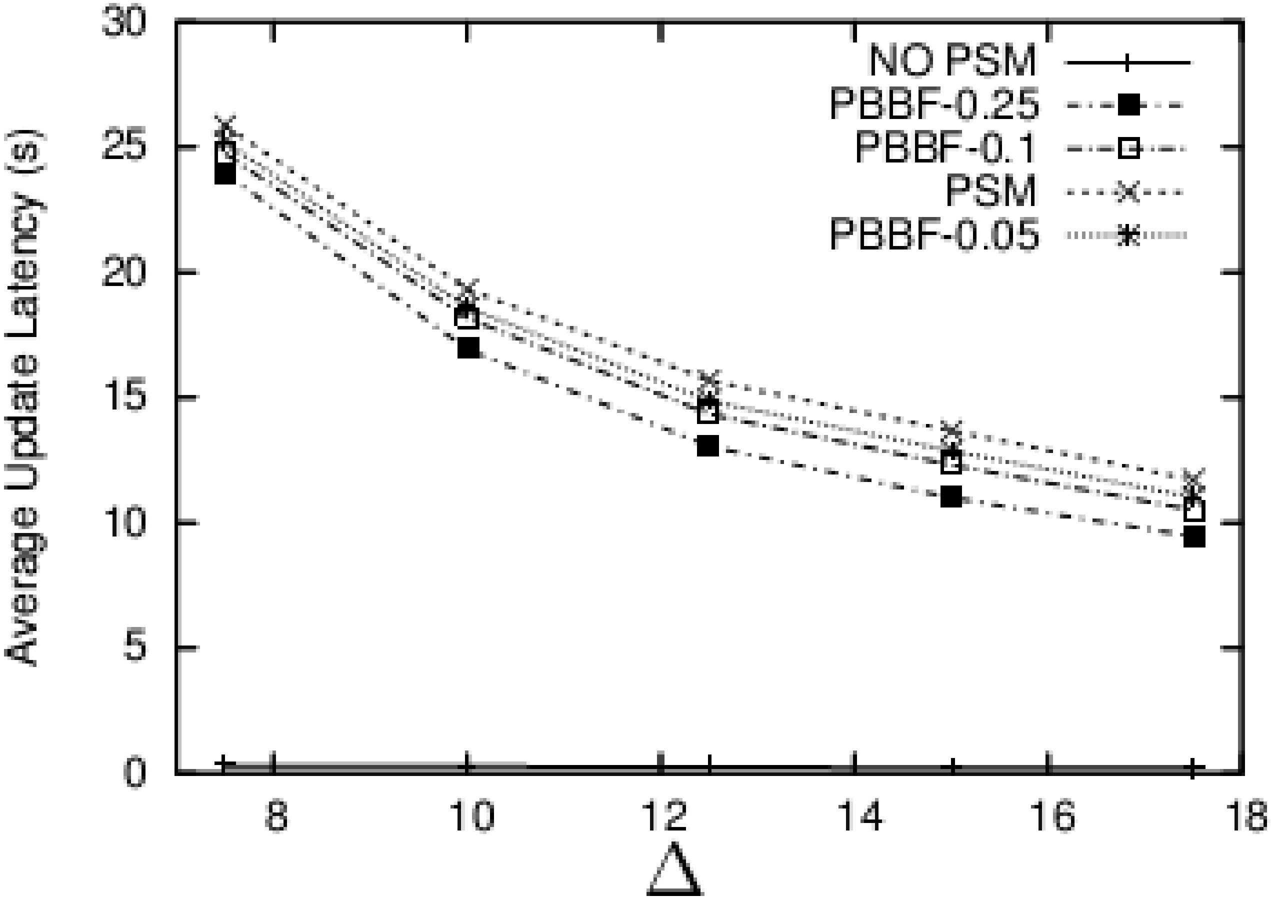



Average energy consumption.




2-hop average update latency.

# Impact of 'Δ'



NO PSM
PBBF-0.25
PBBF-0.1
PSM
PBBF-0.05
Average Update Latency (s)
0
5
10
15
20
25
30
8
10
12
14
16
18
Δ

# Discussions

- Why the simulation graphs don't contain readings for p = 0.75 and in some, for p = 0.5?
- Can PBBF be adapted for unicast protocols?
- Can p and q be decided dynamically for optimization instead of developer setting it?
  - **Adaptive probability-based broadcast forwarding in energy-saving sensor networks** **,** Journal, ACM Transactions on Sensor Networks (TOSN), Volume 4 Issue 2, March 2008, Article No. 6
- Can this mechanism be extended to take advantage of the knowledge of power available at a node or the nodes view of its neighbors?
- Experiments only cover grid-network topology
- Individual nodes in the network cannot currently be configured to have different p and q values