Practical Exploitation of the Energy-Latency Tradeoff for Sensor Network Broadcast

Matthew J. Miller (Cisco Systems)
Indranil Gupta (Univ. of Illinois-Urbana)

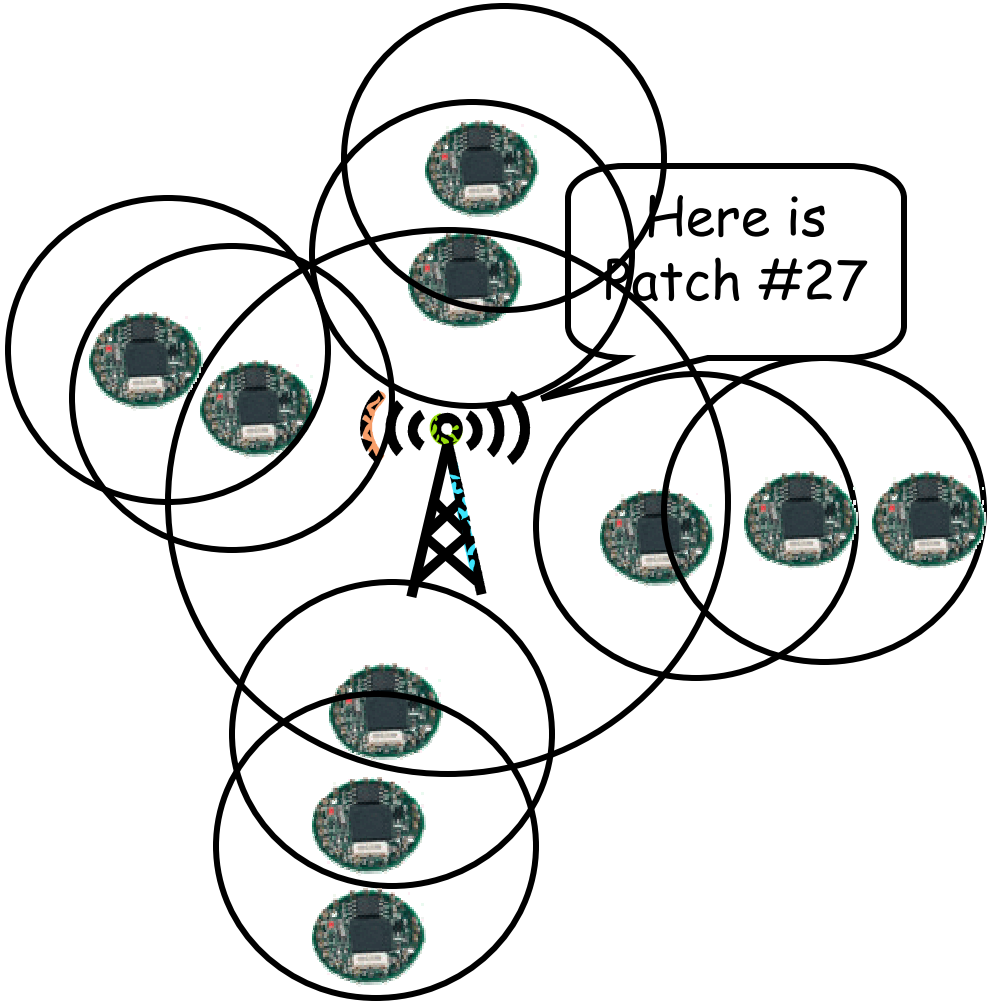
March 2007

Motes: State of the Art

- TinyOS includes a power save protocol (B-MAC)
- Other add-on protocols are implemented (e.g., S-MAC)
- Do these give broadcast applications sufficient flexibility for energy, latency, and reliability metrics?

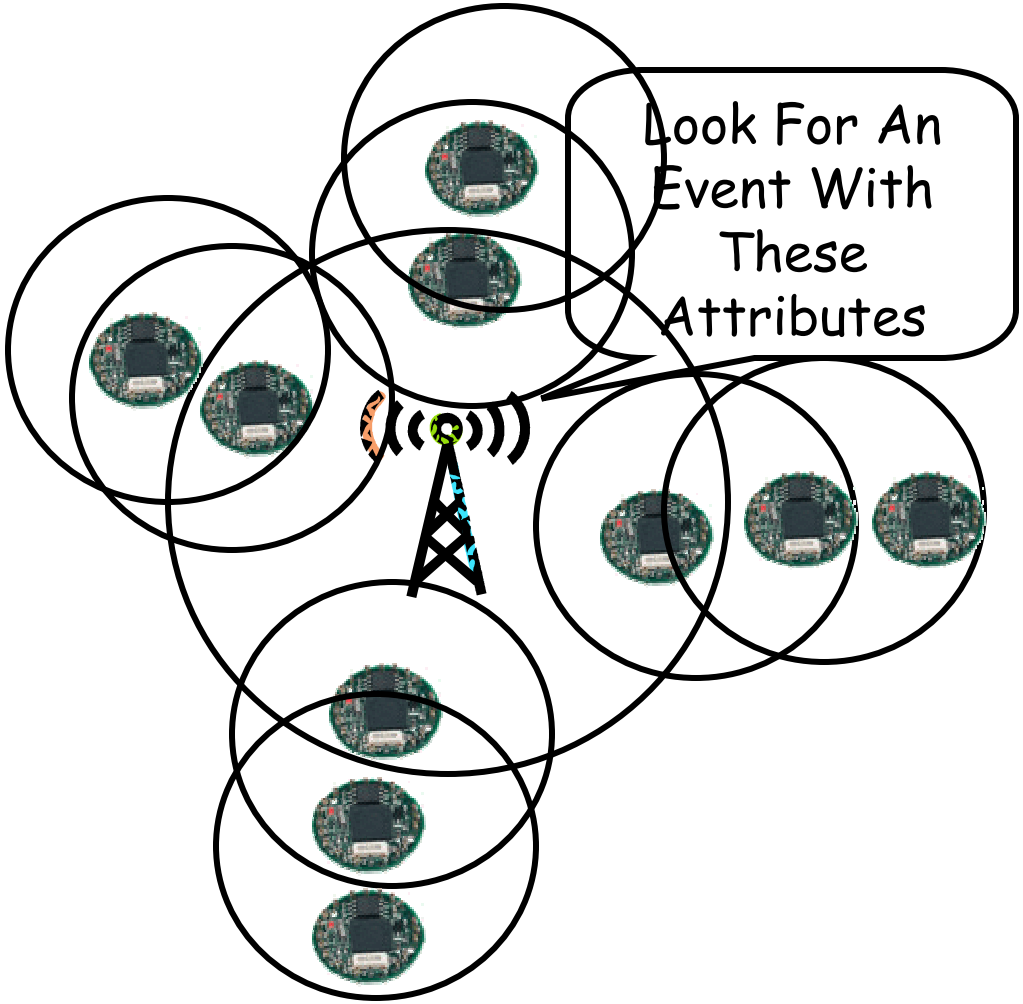
Sensor Application #1

- Code Update Application
  - E.g., Trickle [Levis et al., NDSI 2004]
- Updates Generated Once Every Few Weeks
  - Reducing energy consumption is important
  - Latency is not a major concern

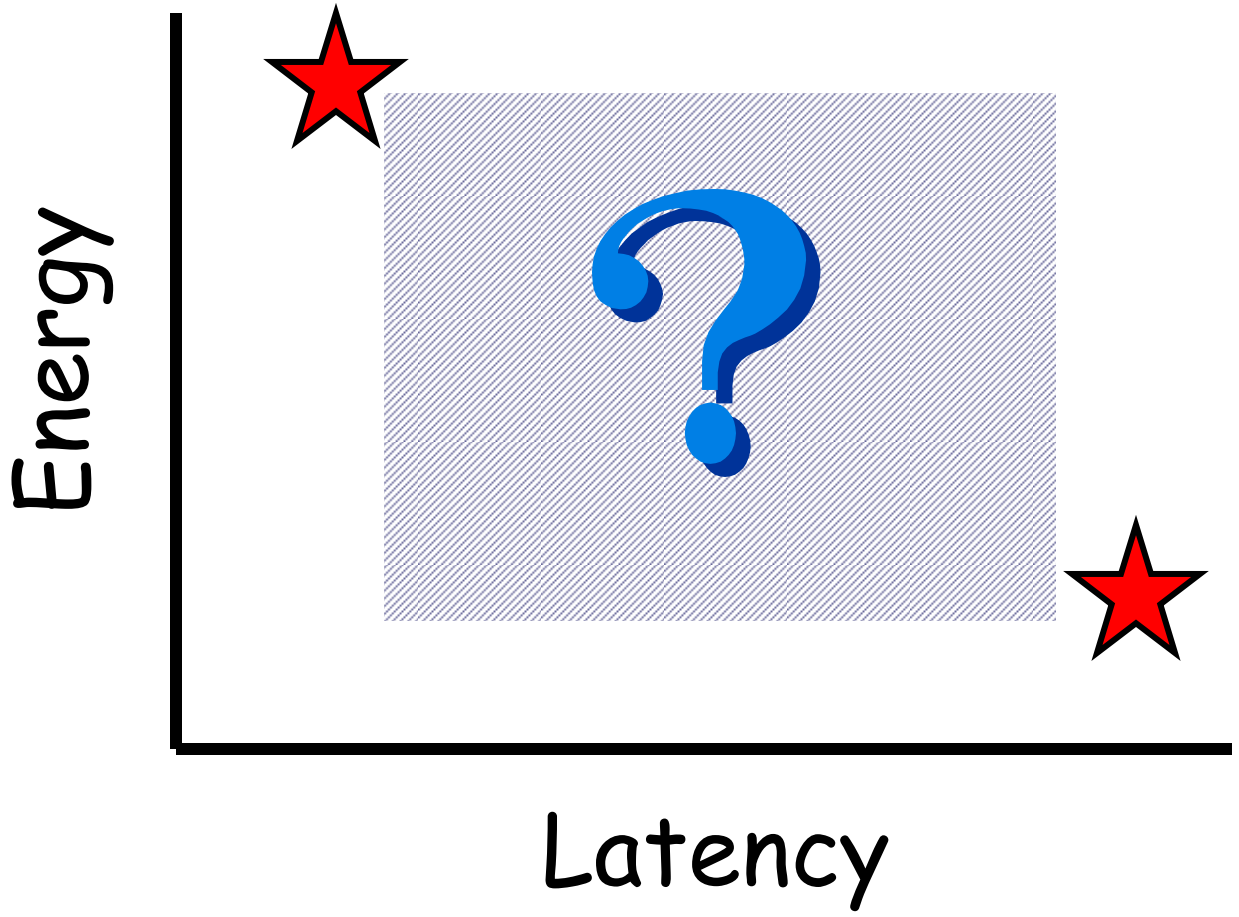


Sensor Application #2

- Short-Term Event Detection
  - E.g., Directed Diffusion [Intanagonwiwat et al., MobiCom 2000]
- Intruder Alert for Temporary, Overnight Camp
  - Latency is critical
  - With adequate power supplies, energy usage is not a concern



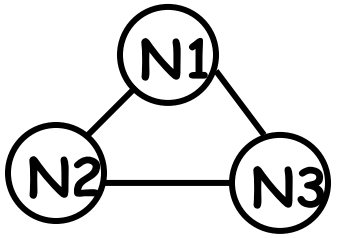
Energy-Latency Options

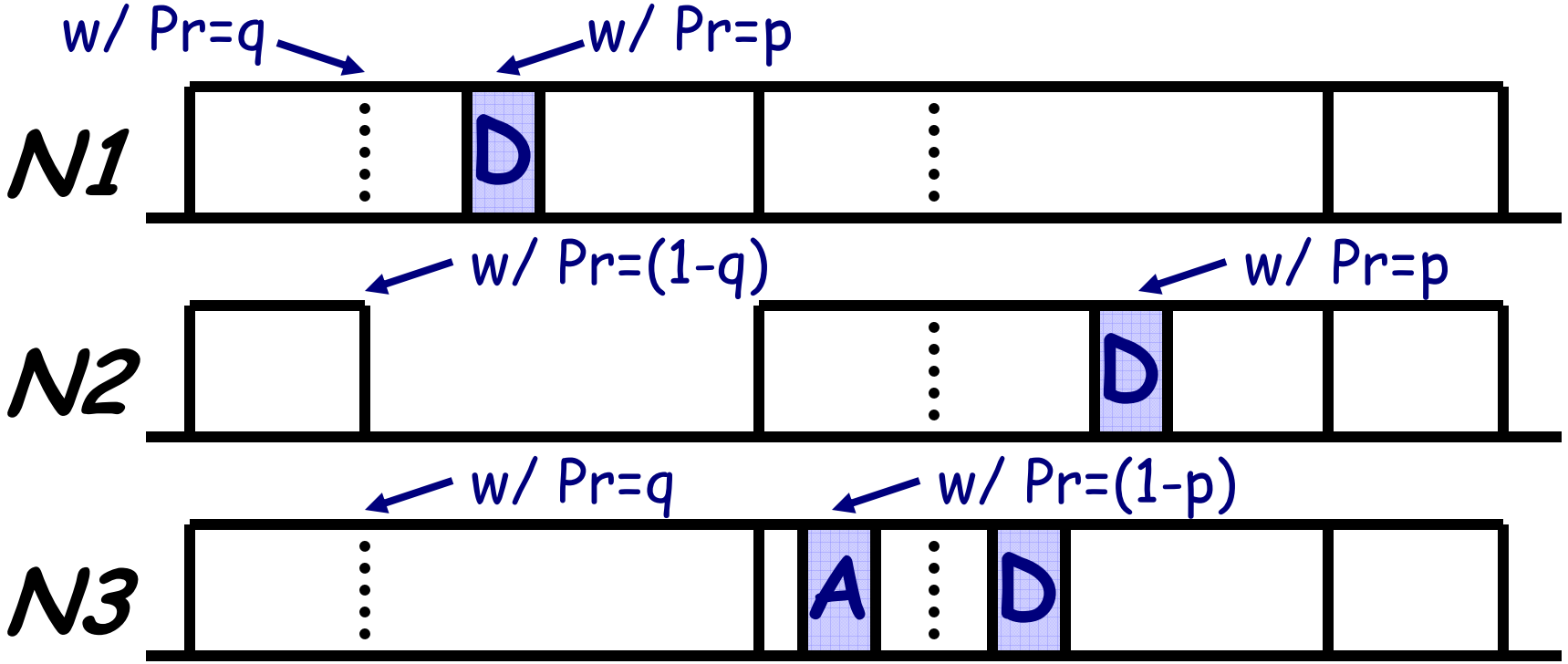


Review of Previous Work

- Probability-Based Broadcast Forwarding (PBBF) provides two probabilistic parameters
  - Allows fine-tuning of the operating point on energy-latency plane
- Can be use with any protocol that:
  - Has scheduled sleep intervals
  - Can awaken neighbors on-demand
  - Can awaken all neighbors at the same time to receive a broadcast
- Such protocols include B-MAC and 802.11 PSM

Probabilistic Protocol





A = Advertisement Pkt

D = Data Pkt

Probability-Based Broadcast Forwarding (PBBF)

- Introduce two parameters to sleep scheduling protocols: p and q
- When a node is scheduled to sleep, it will remain active with probability q
- When a node receives a broadcast, it sends it immediately with probability p
  - With probability (1-p), the node will wait and advertise the packet during the next AW before rebroadcasting the packet

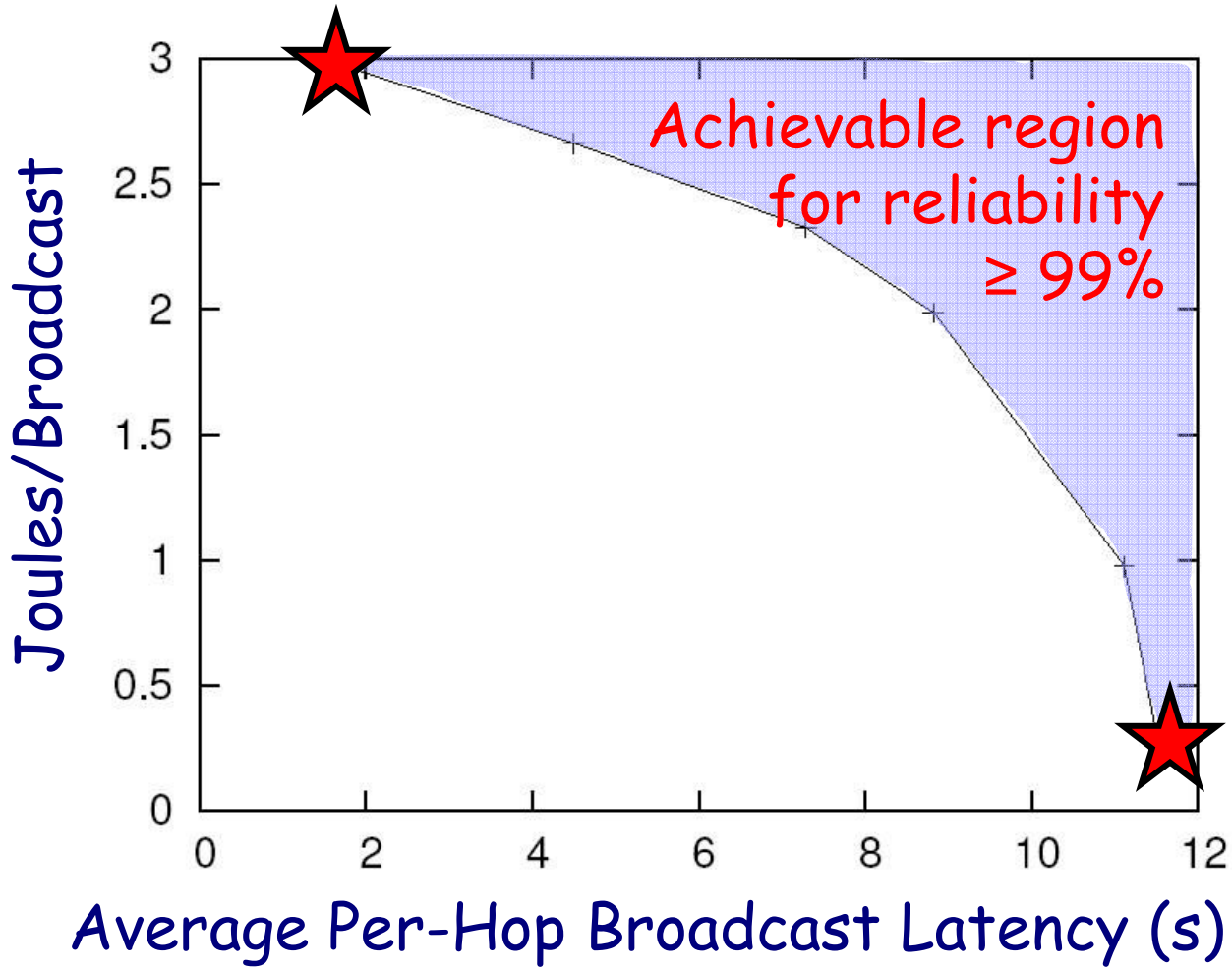
PBBF Comments

- p=0, q=0 equivalent to the original power save protocol
- p=1, q=1 approximates the "always on" protocol
  - Still have the advertisement window overhead

- Effects of p and q on metrics:

	Energy	Latency	Reliability
p ↑ (Immediate Send)	---	↓ if q > 0	↓ if q > 0
q ↑ (Stay On)	↑	↓ if p > 0	↑ if p > 0

Energy-Latency Tradeoff

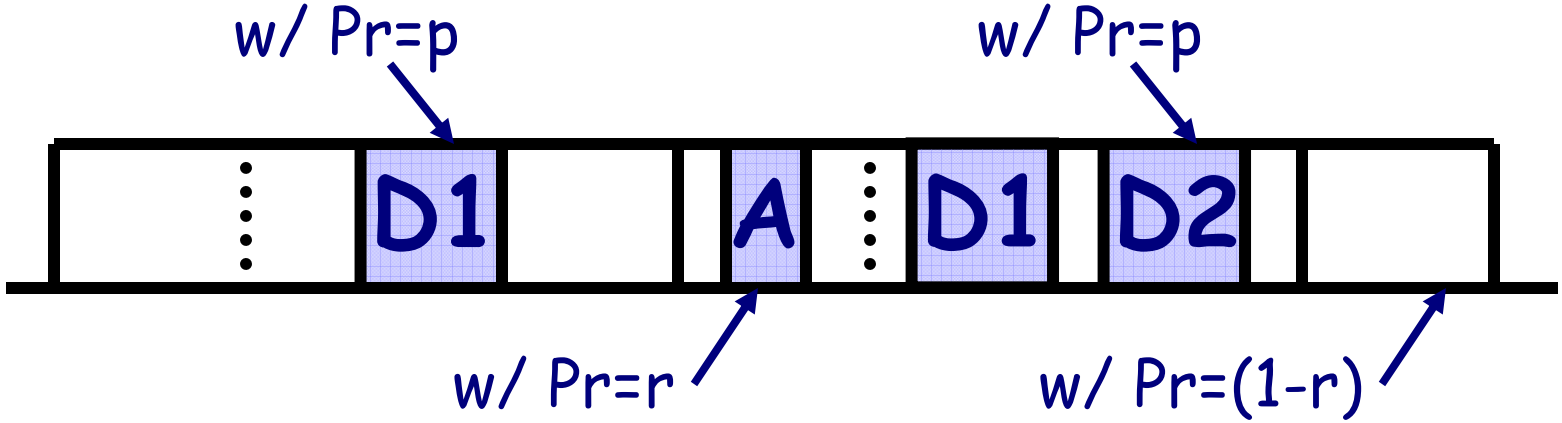


Contributions in This Paper

- Add a new extension to PBBF for more control over reliability and packet overhead
- Implement PBBF in TinyOS on top of B-MAC
  - Proof-of-concept that validates previous work's simulation trends
  - Demonstrates versatility of PBBF
    - Previous work used 802.11 PSM as the base protocol

PBBF Extension: r parameter

If a broadcast is sent immediately then, with probability r, the packet is sent a second time according to the original power save protocol



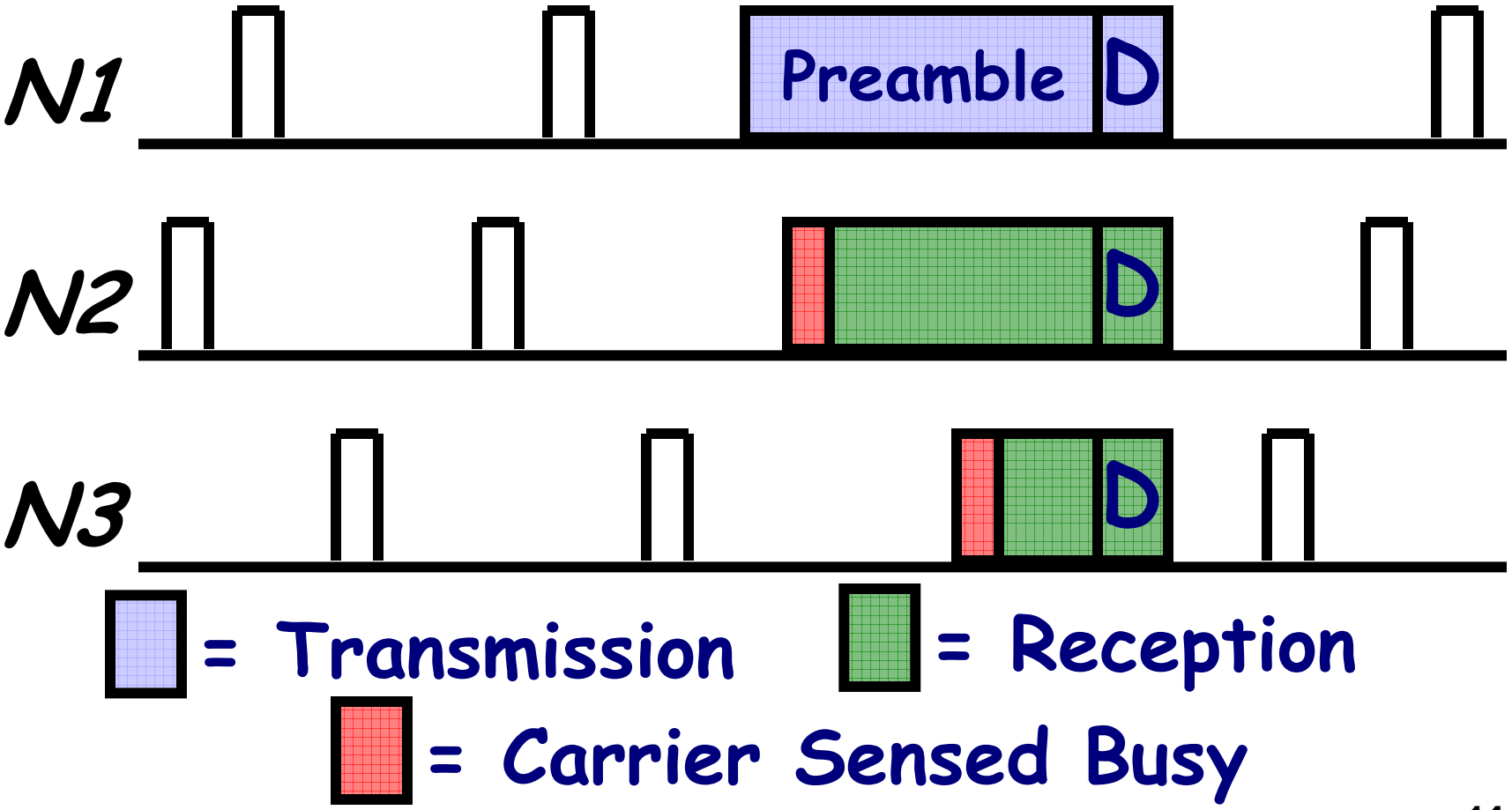
Effects of r Parameter

- Improves reliability, but increases energy consumption, latency, and overhead

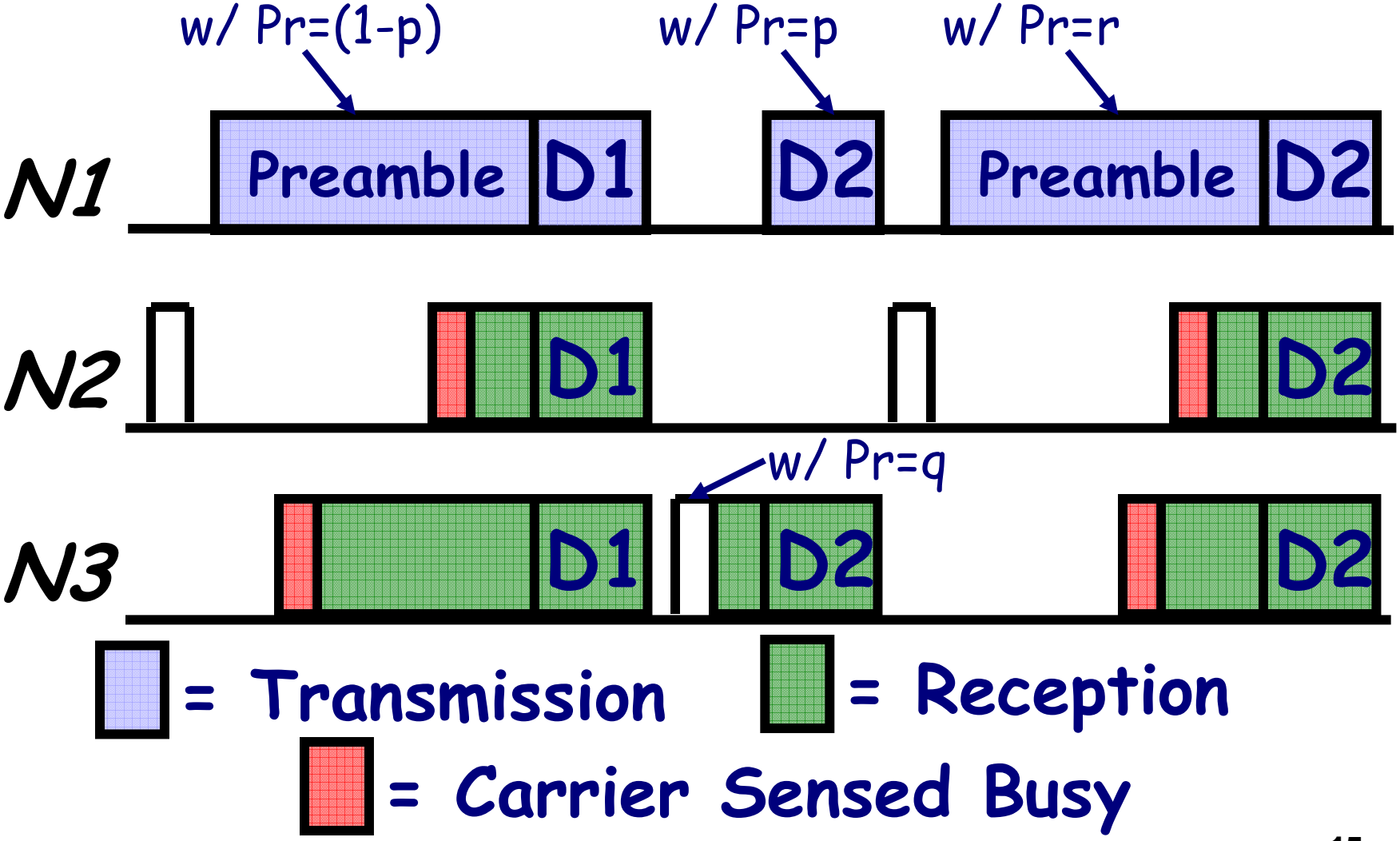
	Energy	Latency	Reliability	Overhead
p ↑ if r = 0	---	↓ if q > 0	↓ if q < 1	---
q ↑ if r = 0	↑	↓ if p > 0	↑ if p > 0	---
r ↑ if p > 0	↑	↑	↑	↑

B-MAC [Polastre et al., SenSys 2004]

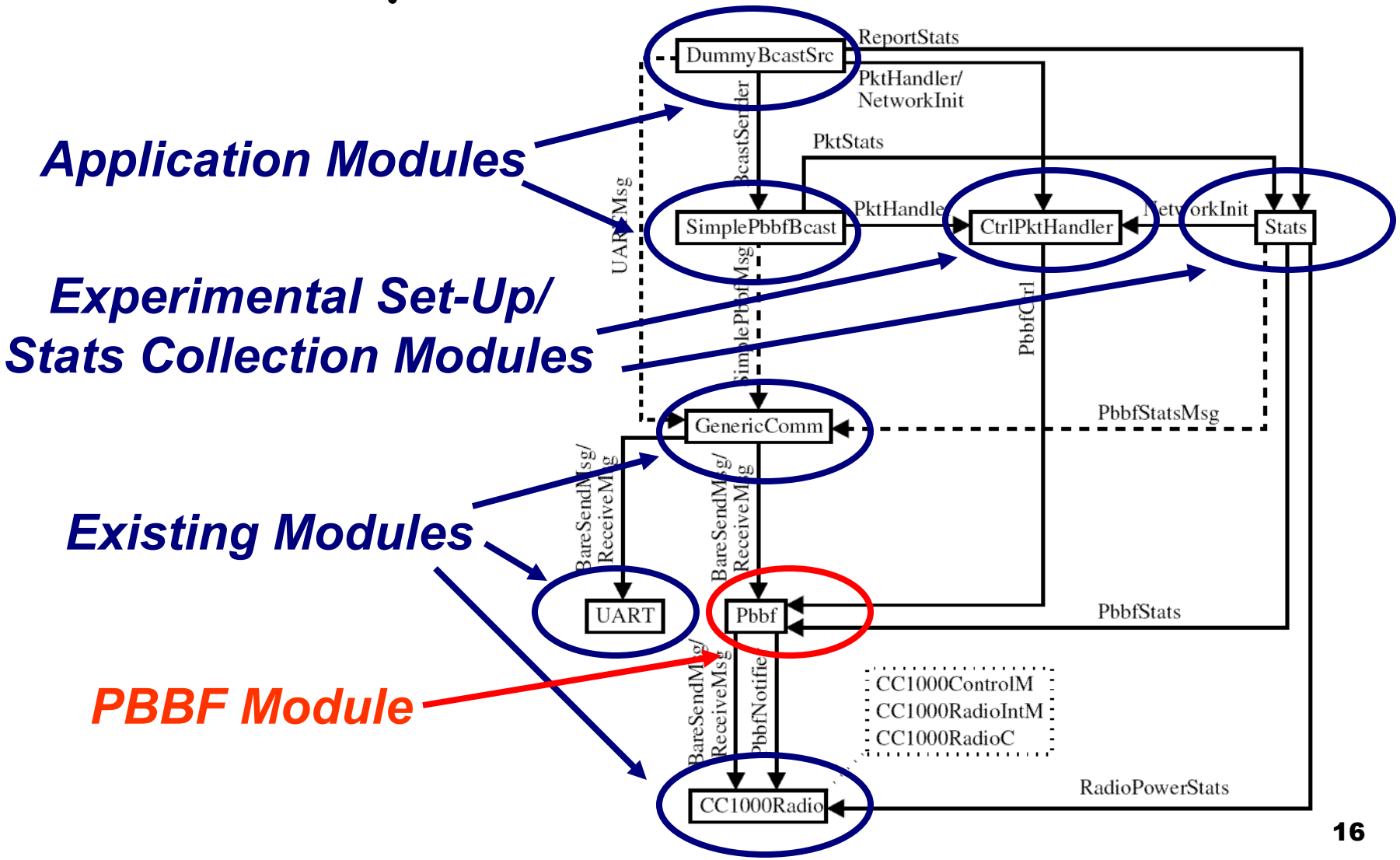
Uses Preamble Sampling



B-MAC with PBBF



Our TinyOS Architecture

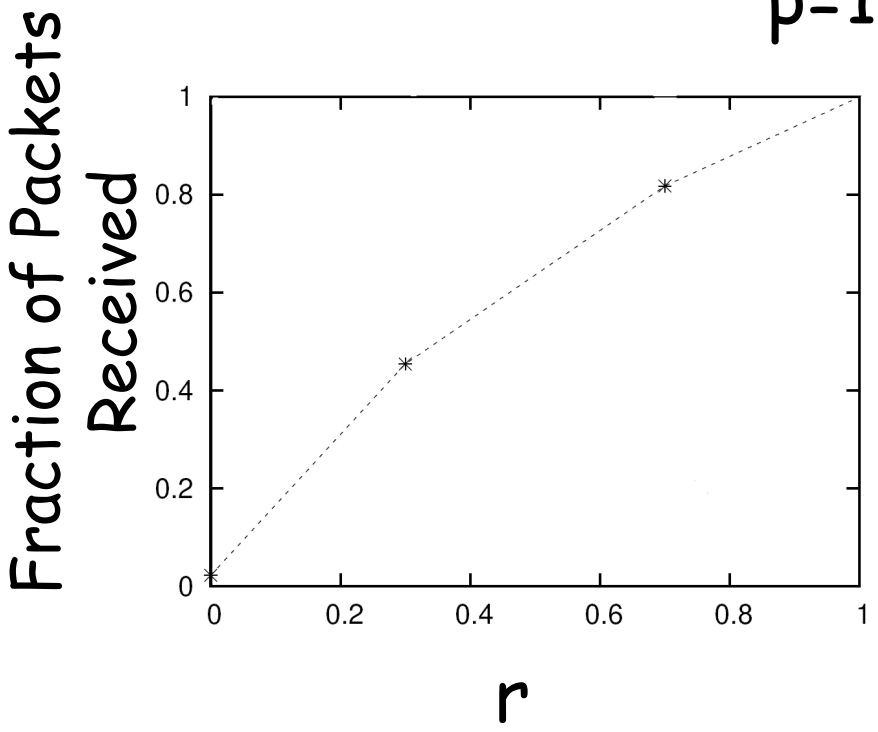


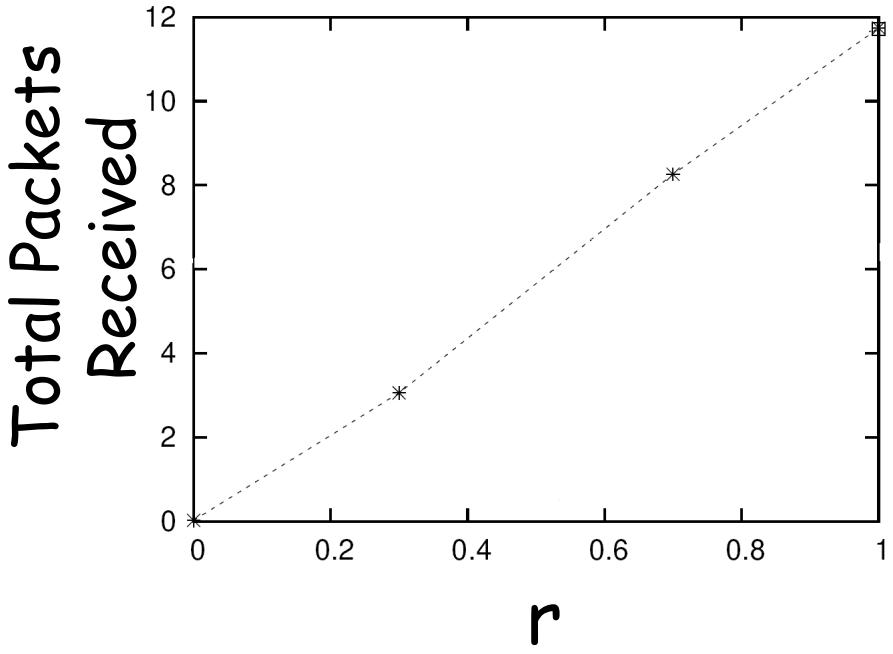
Experimental Setup

- Mica 2 Motes (433 MHz) with one broadcast sender and eight one-hop neighbors
- Broadcast sent every 2.5 s during a 30 s run
- B-MAC parameters
  - Duty cycle = 135 ms
  - Preamble = 371 bytes (155 ms @ 19.2 kbps)
  - Carrier sense time = 8 ms

Experimental Results

p=1 and q=0

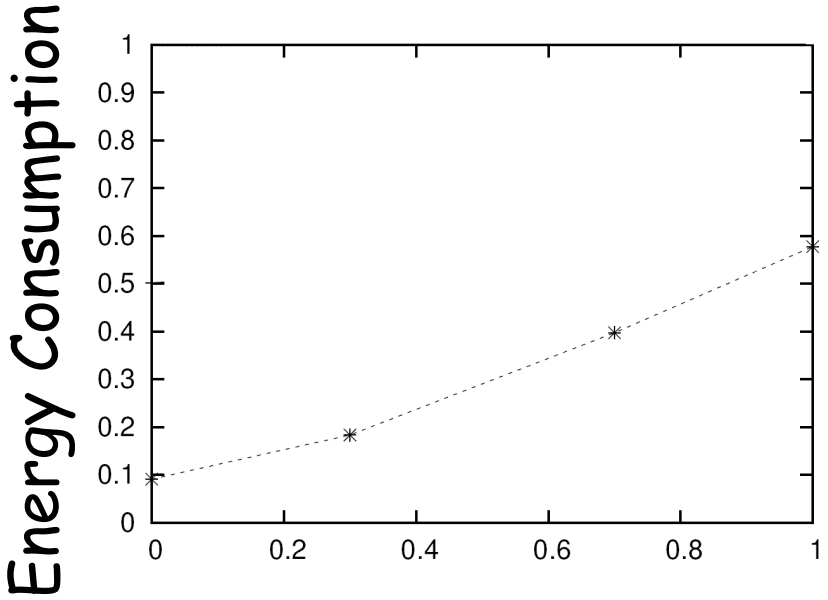


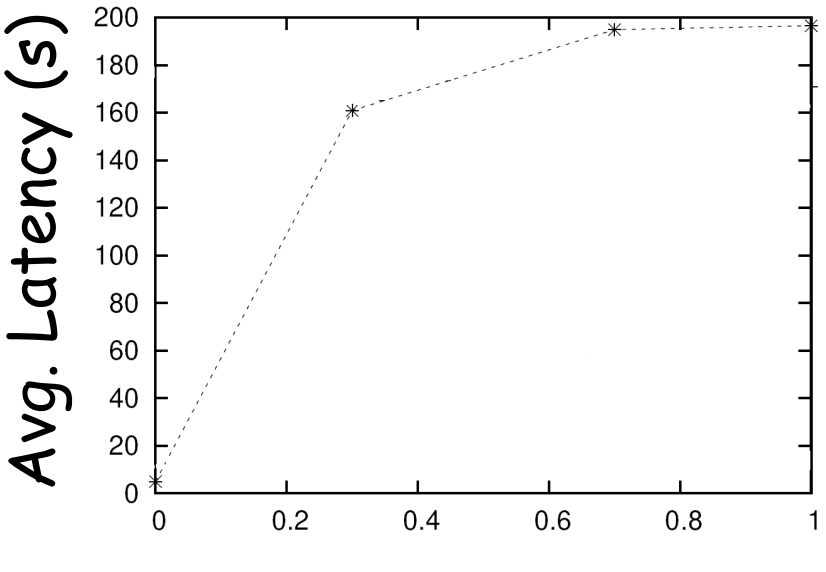


As r increases: reliability increases but so does packet overhead

Experimental Results

p=1 and q=0





As r increases: energy consumption and latency increase (tradeoff for improved reliability)

Conclusion

- We have implemented PBBF for TinyOS
  - Code can be downloaded at:
    www.matthewjmiller.net/code
- Shown versatility of PBBF concept
  - Can be used with B-MAC in addition to 802.11 PSM
- New parameter can adjust reliability
  - Improved reliability comes at the cost of degradation to packet overhead, energy consumption, and latency

Questions???

Matthew J. Miller

www.matthewjmiller.net

Indranil Gupta

www-faculty.cs.uiuc.edu/~indy