Energy Efficiency and Security for Multihop Wireless Networks

Matthew J. Miller

Final Defense

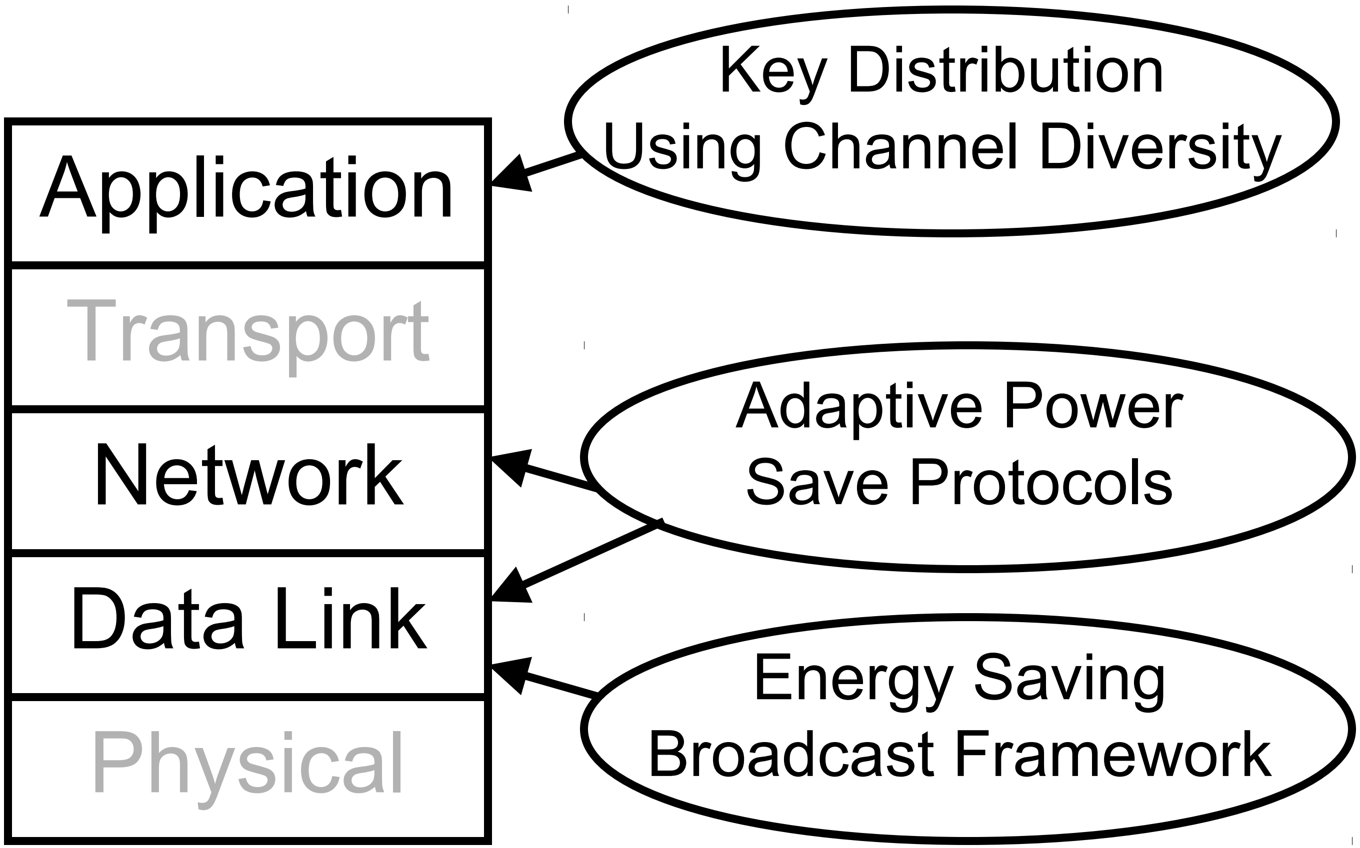
May 31, 2006

Thesis Goals: Energy Efficiency and Security

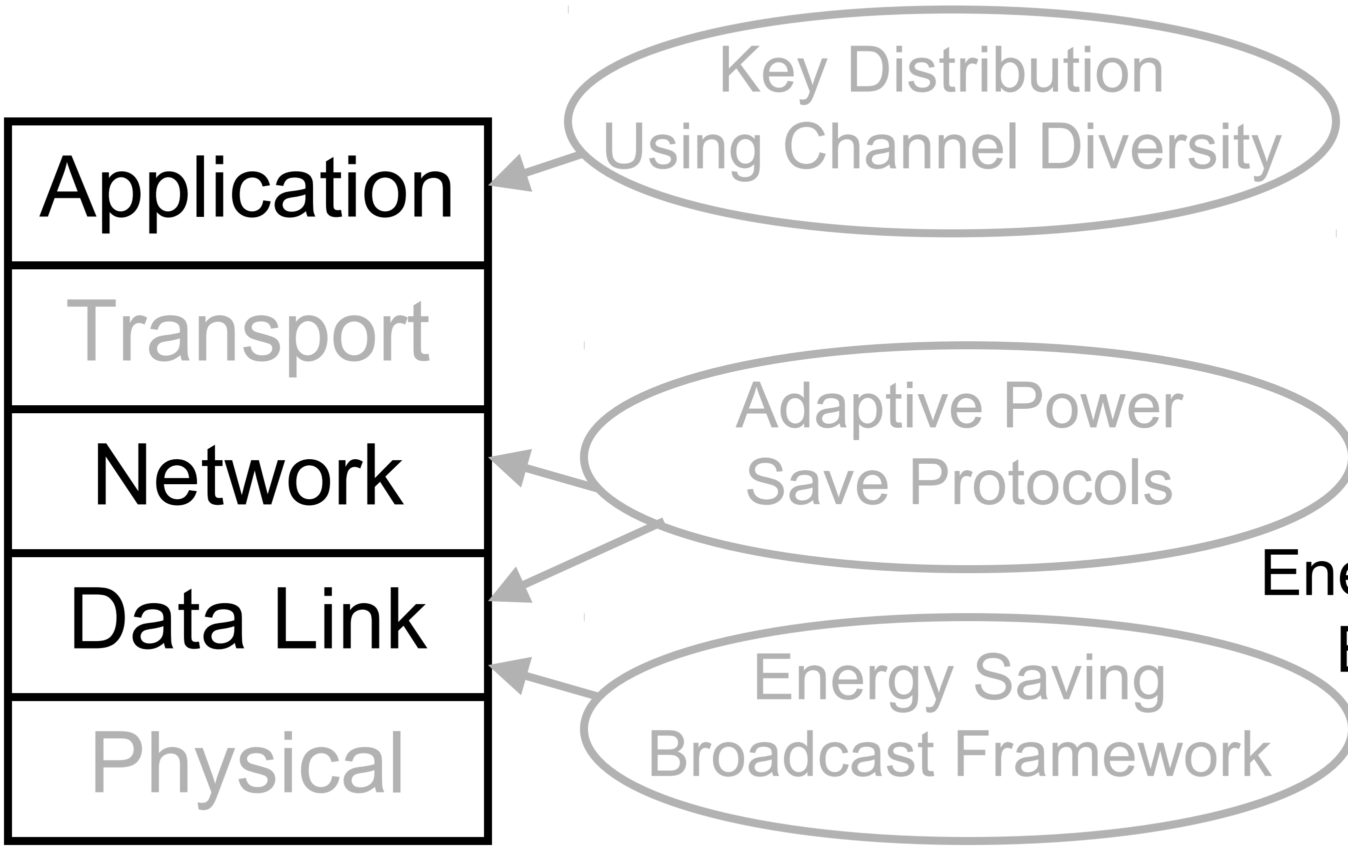
- Both areas need significant improvement for ubiquitous wireless networks to become a reality

- Energy Efficiency: Marginal gains in batteries necessitate power save protocols
- Security: Resource constrained devices with insecure wireless channels

Thesis Outline



Thesis Outline



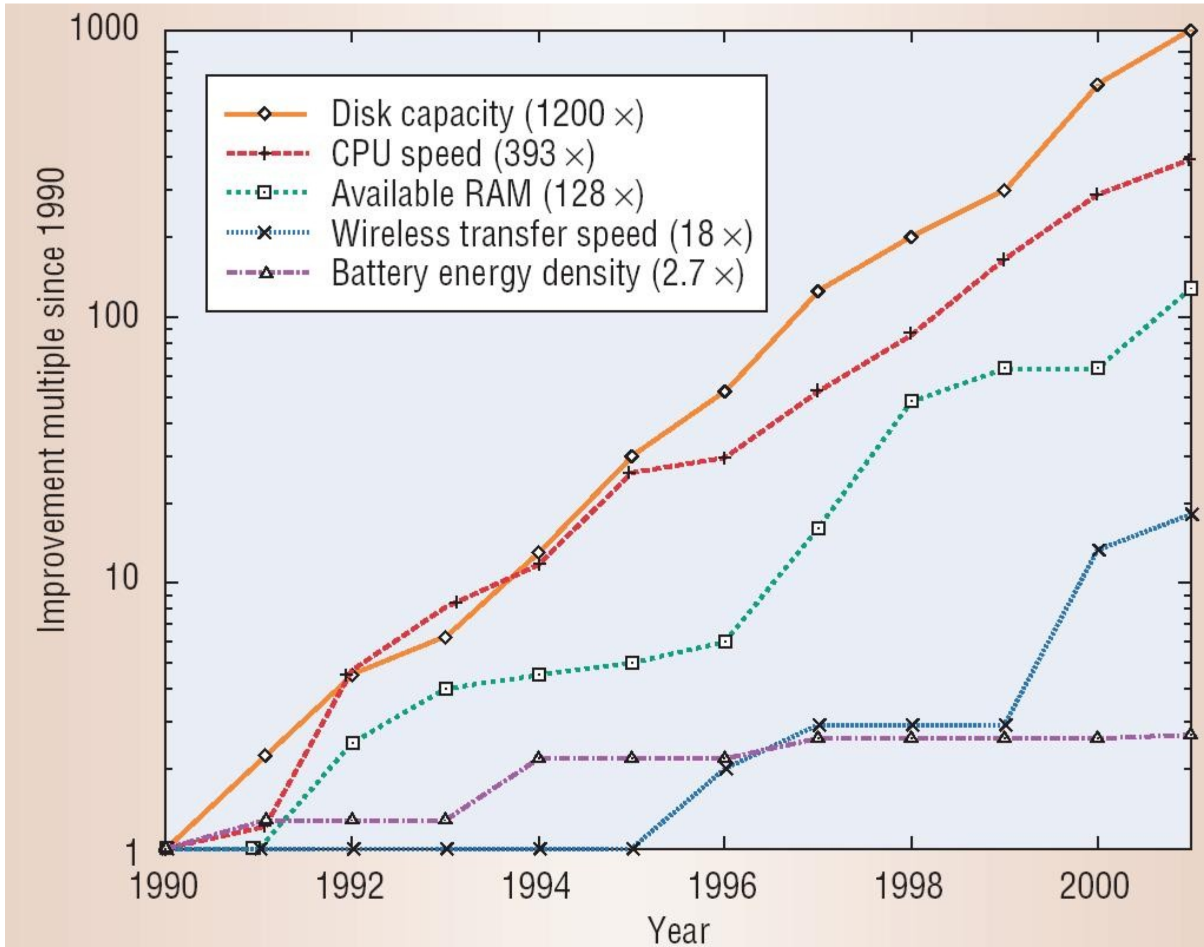
Energy Efficiency
Background

Won’t Moore’s Law Save Us?

NO!!!

Necessitates Energy-Saving Protocol Design

Log Scale



1200 x

393 x

128 x

18 x

2.7 x

From “Thick Clients for Personal Wireless Devices” by Thad Starner in *IEEE Computer*, January 2002

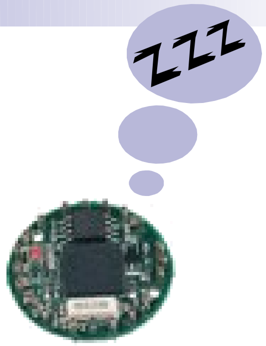
Energy Consumption Breakdown

From UIUC Vodafone Symposium	Data Traffic (Laptop)	Voice Traffic (Cell Phone)
Display	45%	2%
Radio Transmit	5%	24%
Radio Receive/Listen	10%	37%
CPU	40%	37%

Source: Nikhil Jain, Qualcomm

- Solution spans multiple areas of research: networking, OS, architecture, and applications
- Our work focuses on the networking component
- While applicable to laptops, our work is most beneficial to small/no display devices like sensors

How to Save Energy at the Wireless Interface

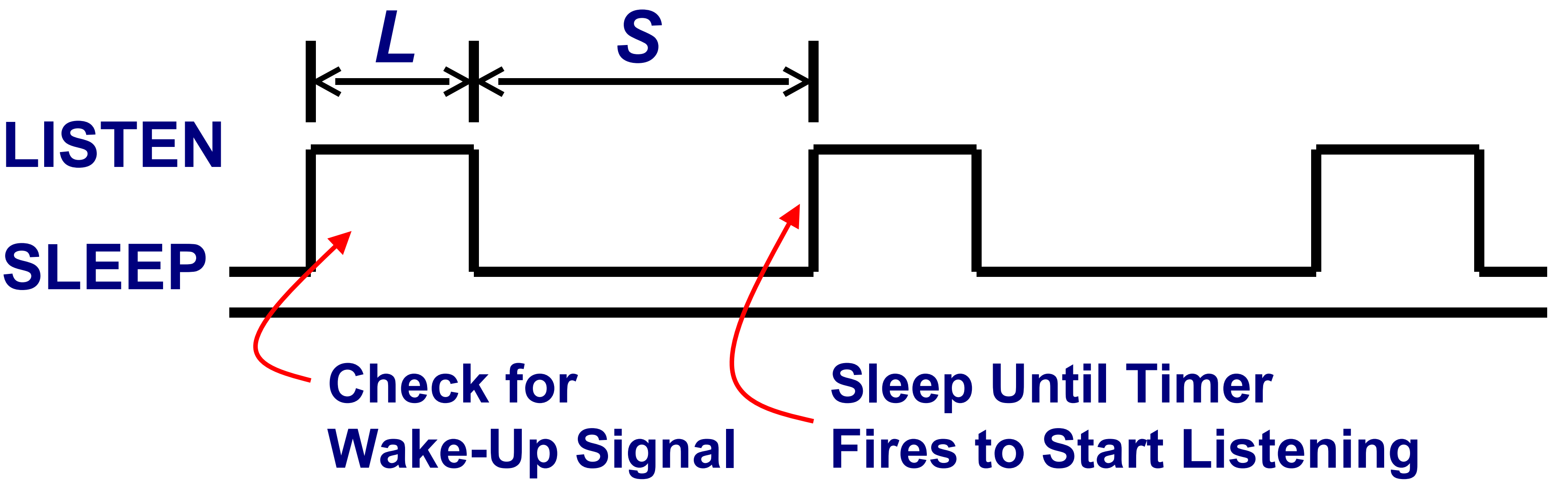


Specs for Mica2 Mote Radio

Radio Mode	Power Consumption (mW)
TX	81
RX/Idle	30
Sleep	0.003

- Sleep as much as possible!
- Fundamental Question: *When should a radio switch to sleep mode and for how long?*
- Must balance energy saving with *latency* needs

Common Power Save Protocol Design



- *L* and *S* are static values regardless of traffic
- Even with no traffic, the node is awake for *L / (L+S)* fraction of the time
- *L* is on the order of the time to receive a packet

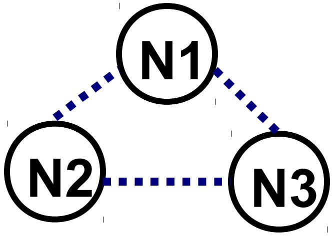
Wake-up Channel Models: In-Band vs. Out-of-Band

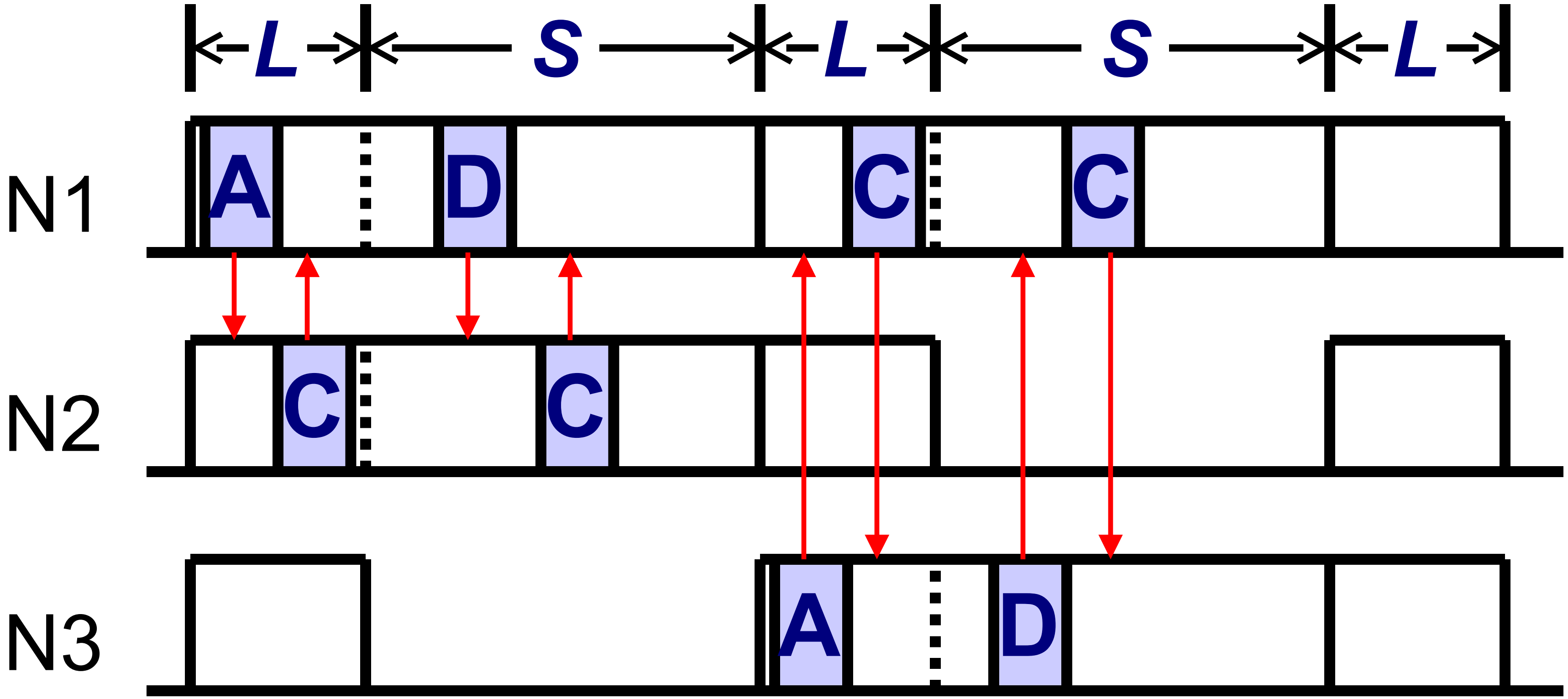
- In-Band
  - Wake-up signaling and data communication use the *same channel*
  - Extra coordination necessary to avoid interference between data packets and wake-up signals
- Out-of-Band
  - Wake-up signaling and data communication use *separate, orthogonal* channels *concurrently*
  - Extra hardware complexity necessary to provide separate, concurrent channels

Protocol Design Space

	Carrier Sensing	Adaptive Listening	Adaptive Sleeping
In-Band	Section 3.1	Section 4.1	Section 4.2
Out-of-Band	Section 3.2	Our in-band techniques are applicable	MS Thesis

In-Band Protocol Example



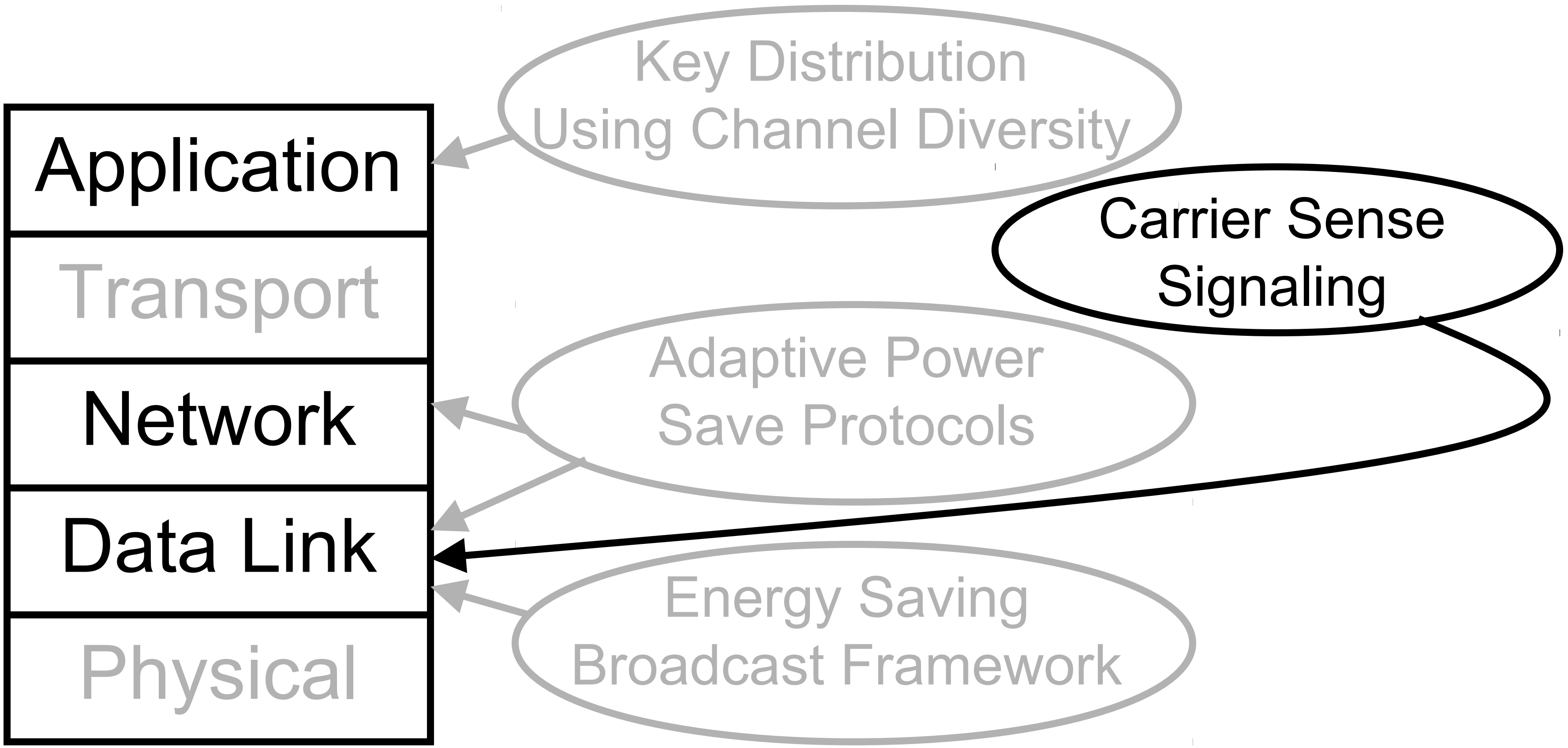


A = Advertisement Pkt

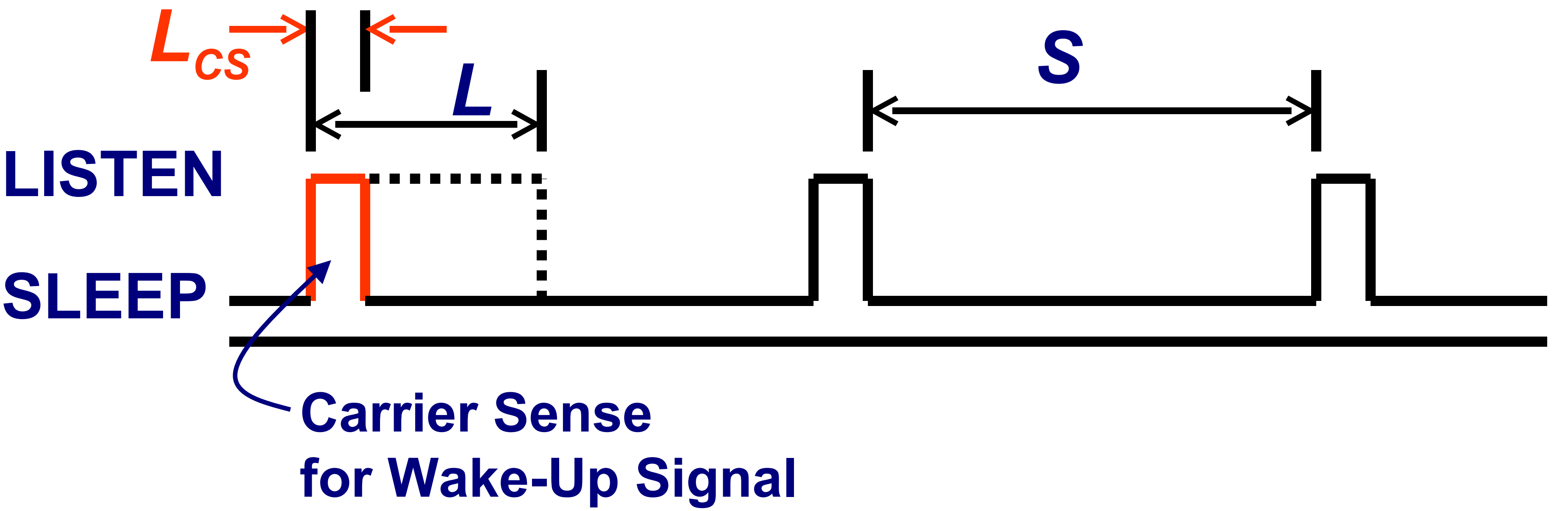
C = ACK Pkt

D = Data Pkt

Thesis Outline

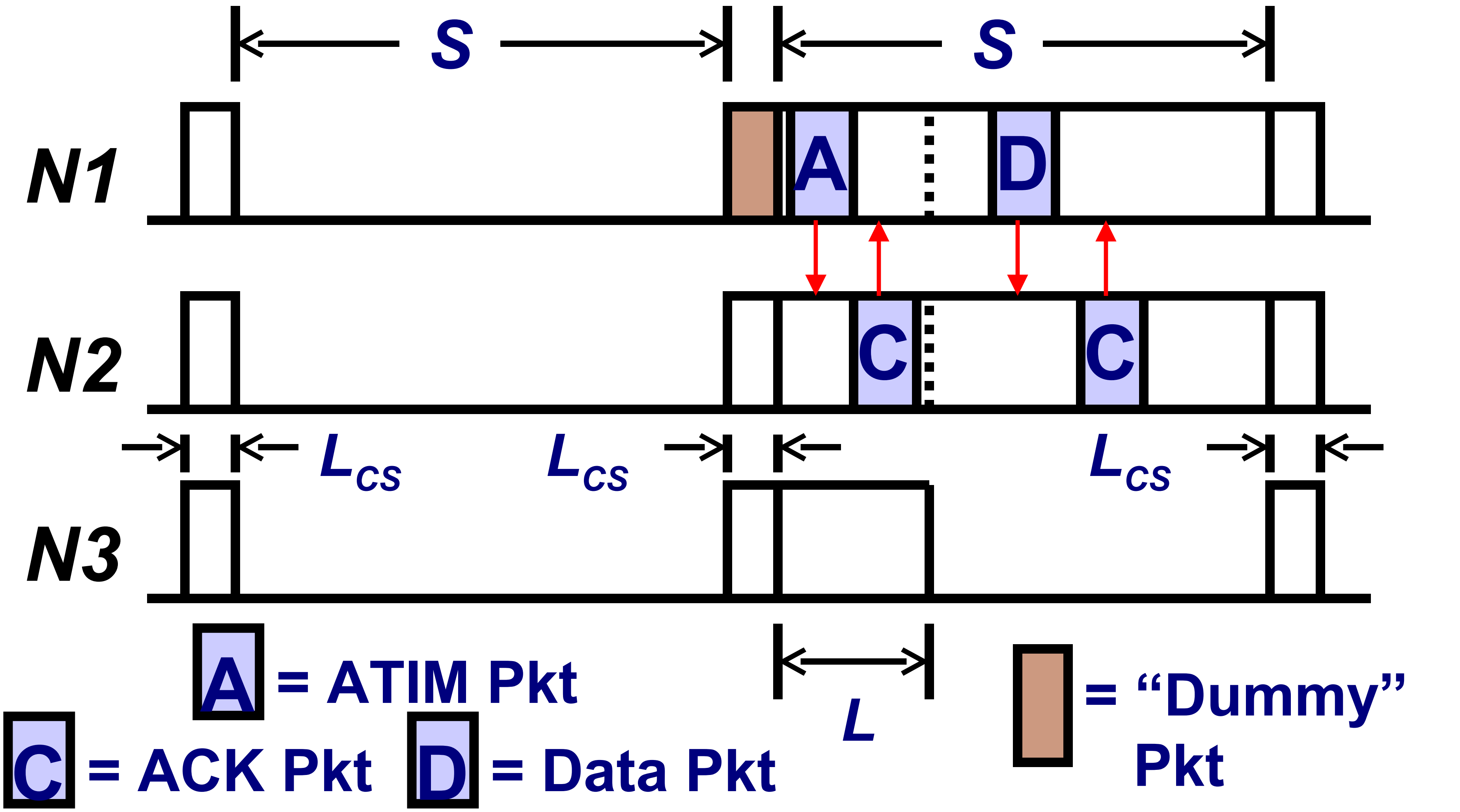


Carrier Sensing for Signaling



- Decrease L to L_{CS} using carrier sensing (CS)
- If carrier is sensed busy, then stay on to receive packet
- Typically, CS time << packet transmission time
  - E.g., 802.11 compliant hardware CS time ≤ 15 μs

Applying CS Signaling to 802.11 PSM



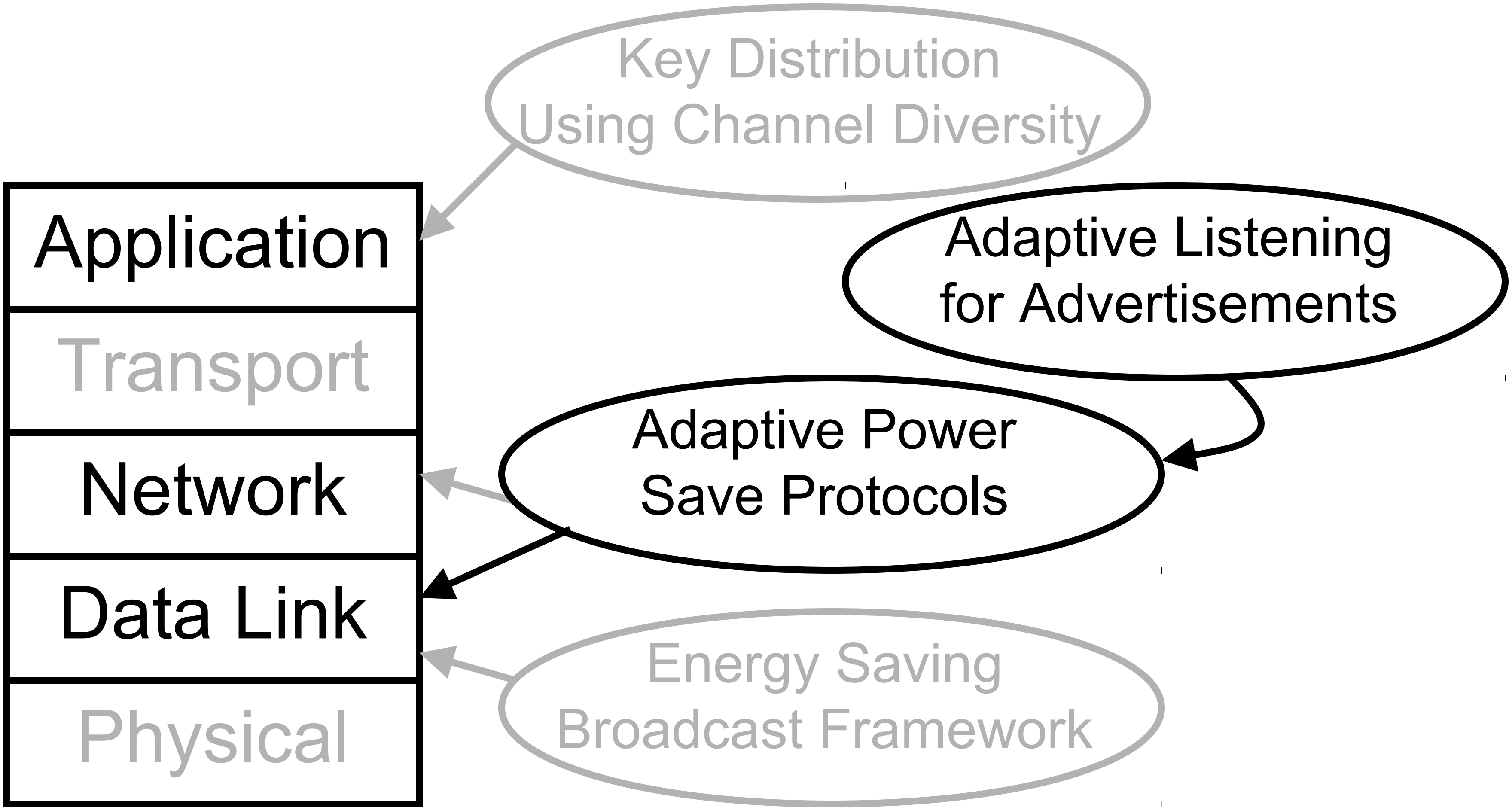
Observations

- When there are no packets to be advertised, nodes use significantly less energy
- Average latency is slightly longer
  - Packets that arrive during the AW are advertised in 802.11 PSM, but may not be with our technique
  - First packet cannot be sent until $L_{CS}+L$ after beginning of BI instead of just L
- False positives may occur when nodes carrier sense the channel busy due to interference
- Can be adapted to other types of power save protocols (e.g., TDMA)

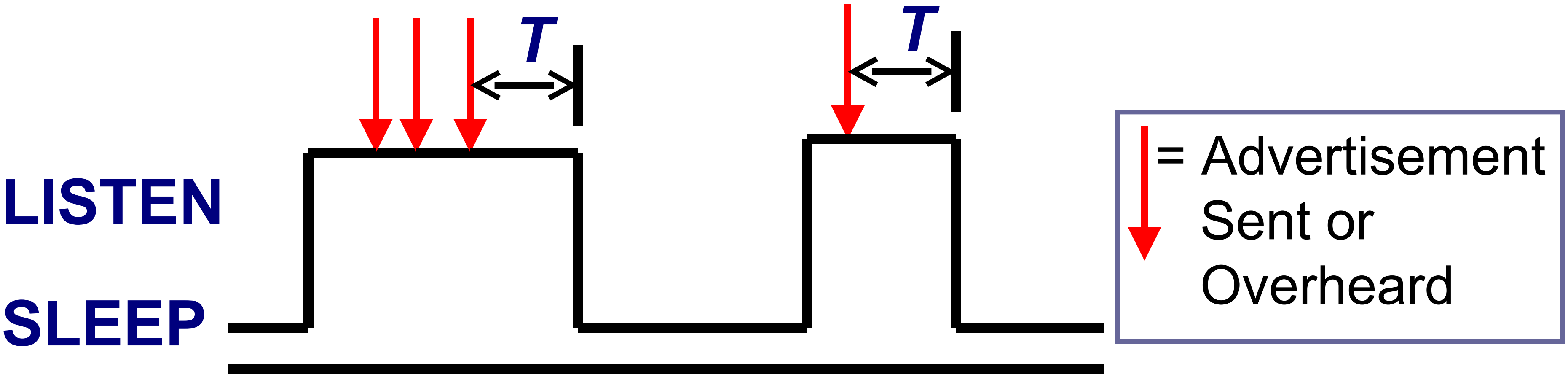
Other Notes

- Results are presented in the next section
  - Carrier sense signaling is combined with adaptive listening
- In Section 3.2, we propose and evaluate carrier sense signaling applied to out-of-band protocols
  - For brevity, we omit a discussion in this presentation

Thesis Outline



Our Approach to Adaptive Listening

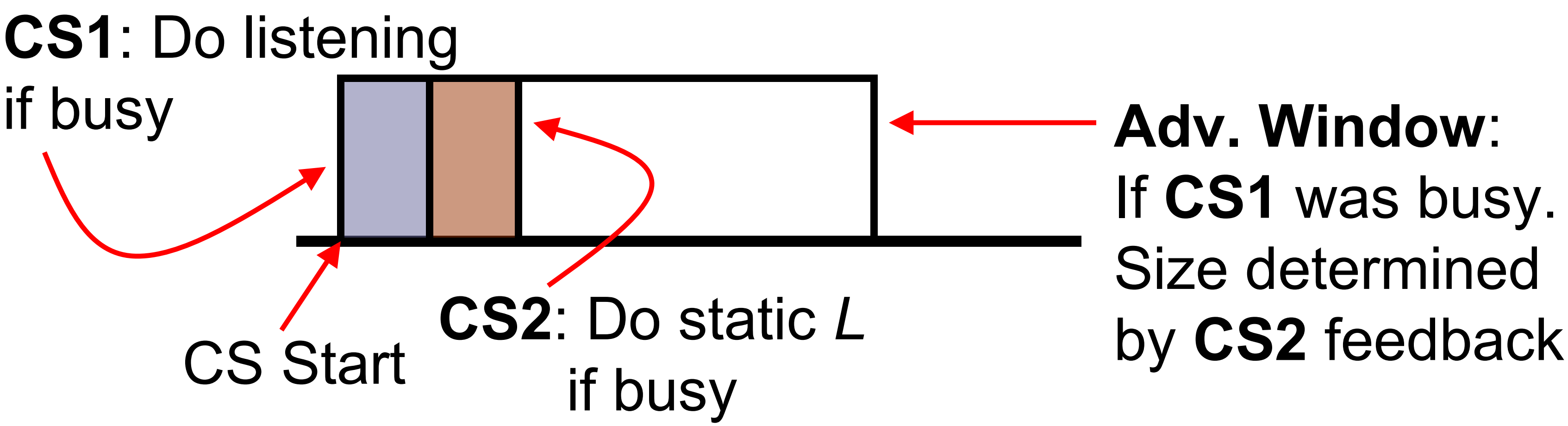


- Use carrier sensing to extend the listening period for advertisements
- Previous work has proposed dynamic listening periods for 802.11 power save, but ours is the first for single radio devices in multihop networks

Adaptive Listening Overview

- Use received signal strength to extend listening as long as a neighbor might try to transmit
- Continue extension as long as sufficiently strong signals are received or a specified upper bound is reached
- Details covered in prelim presentation and thesis

Adaptive Listening and Carrier Sensing



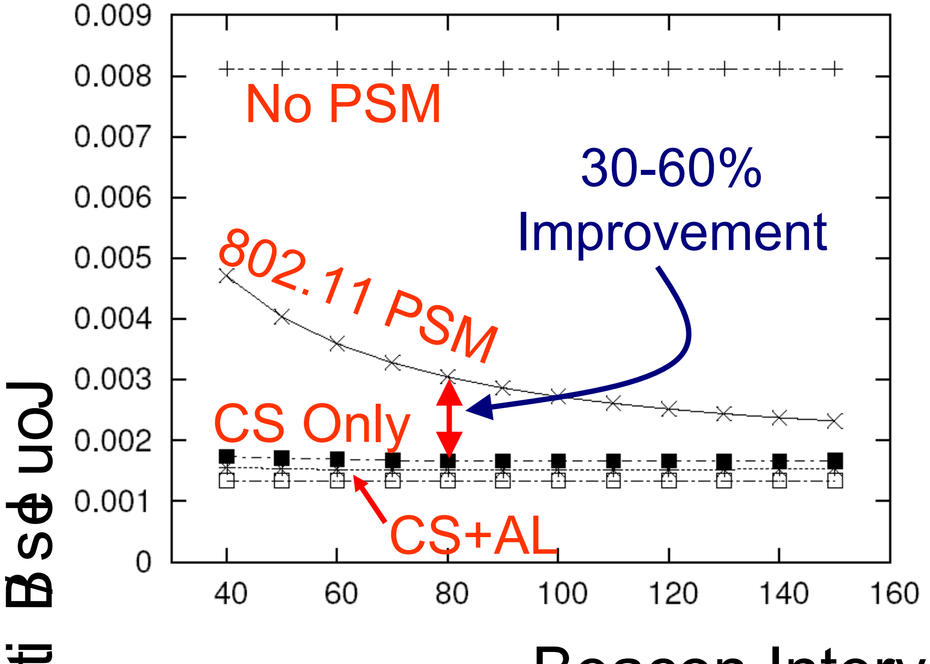
- First CS period indicates whether advertisement window is necessary
- Second CS period indicates whether window size should be fixed or adaptive
  - If a sender repeatedly fails using adaptive listening, it can fallback to the original protocol

Adaptive Listening Results

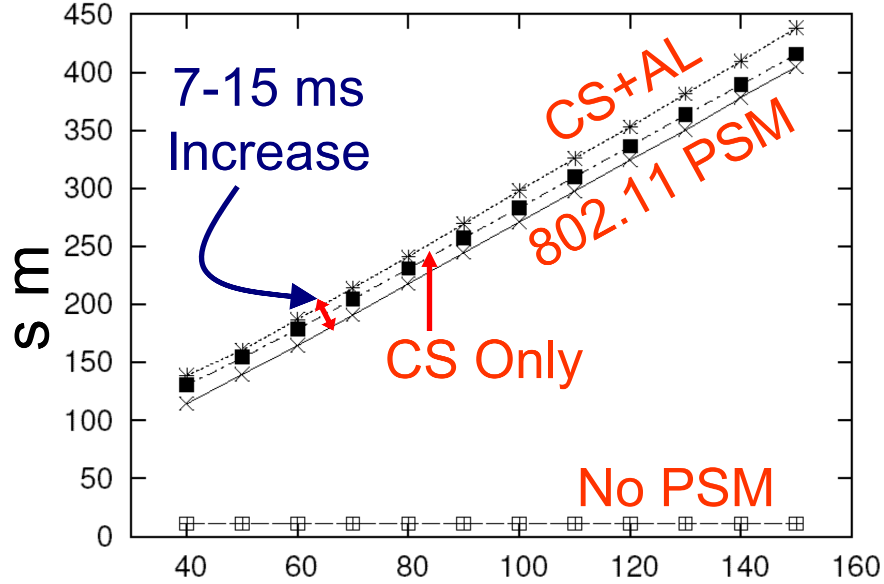
- Simulated using *ns*-2
- Five flows with source and destination selected uniformly at random
  - Lower traffic = 1 kbps per flow
  - Higher traffic = 10 kbps per flow
- CS Only = Carrier sense signaling at beginning of advertisement window only
- CS+AL = Carrier sense signaling at beginning plus adaptive listening

Summary of Results: Lower Traffic

Energy



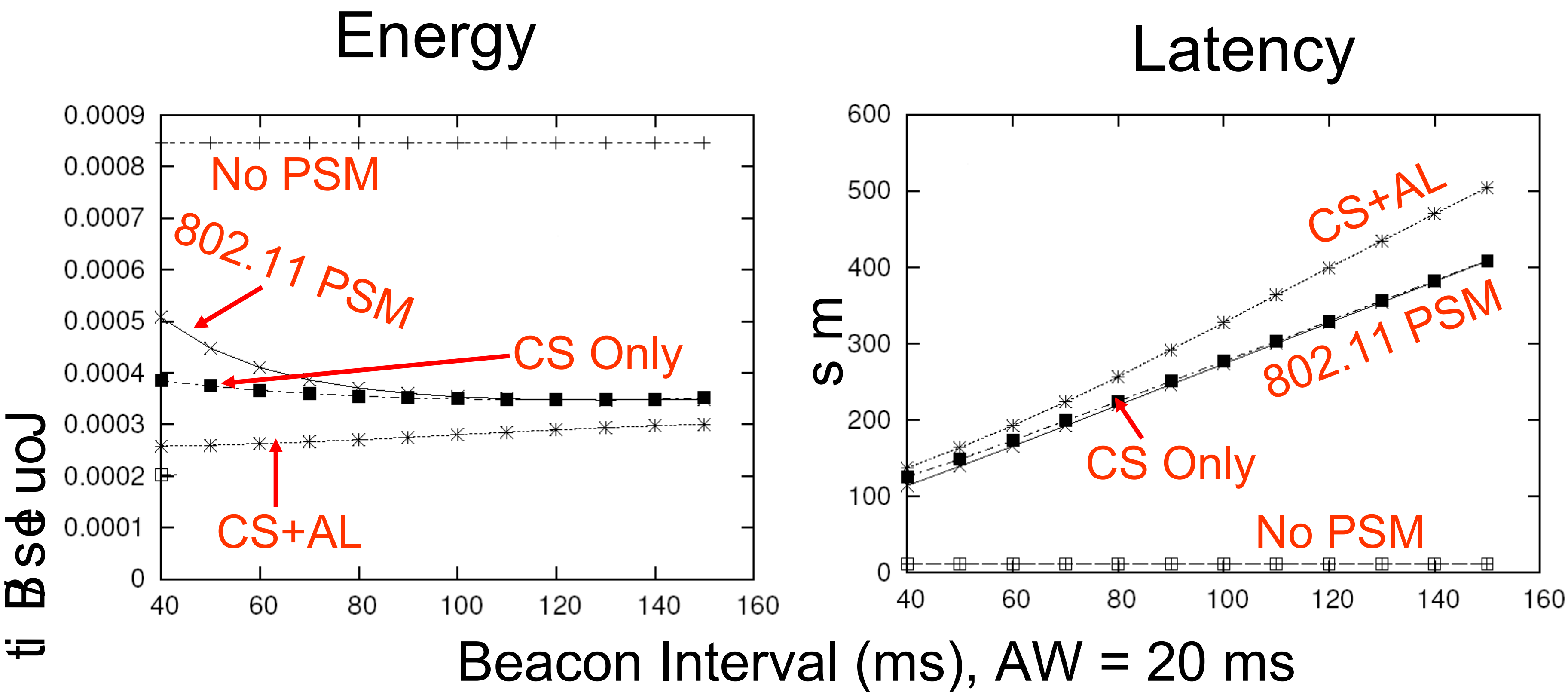
Latency



Beacon Interval (ms), AW = 20 ms

Latency Increase: (1) Additional CS periods, (2) Packets arriving during AW, (3) For adaptive listening, postponed advertisements

Summary of Results: Higher Traffic

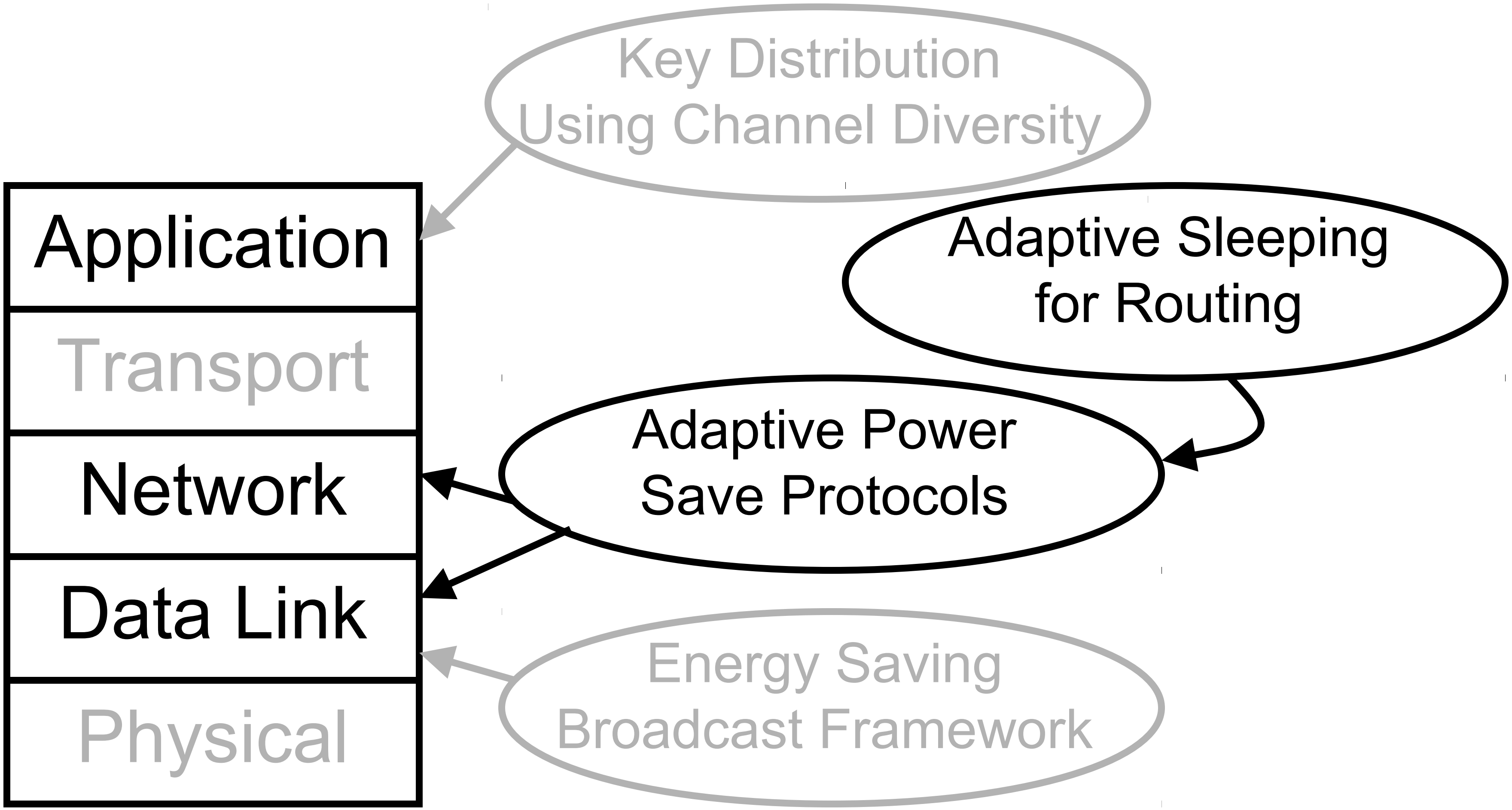


Differences from Lower Traffic: (1) More Adv. windows have at least one packet, (2) More contention means more deferred Advs.

Summary

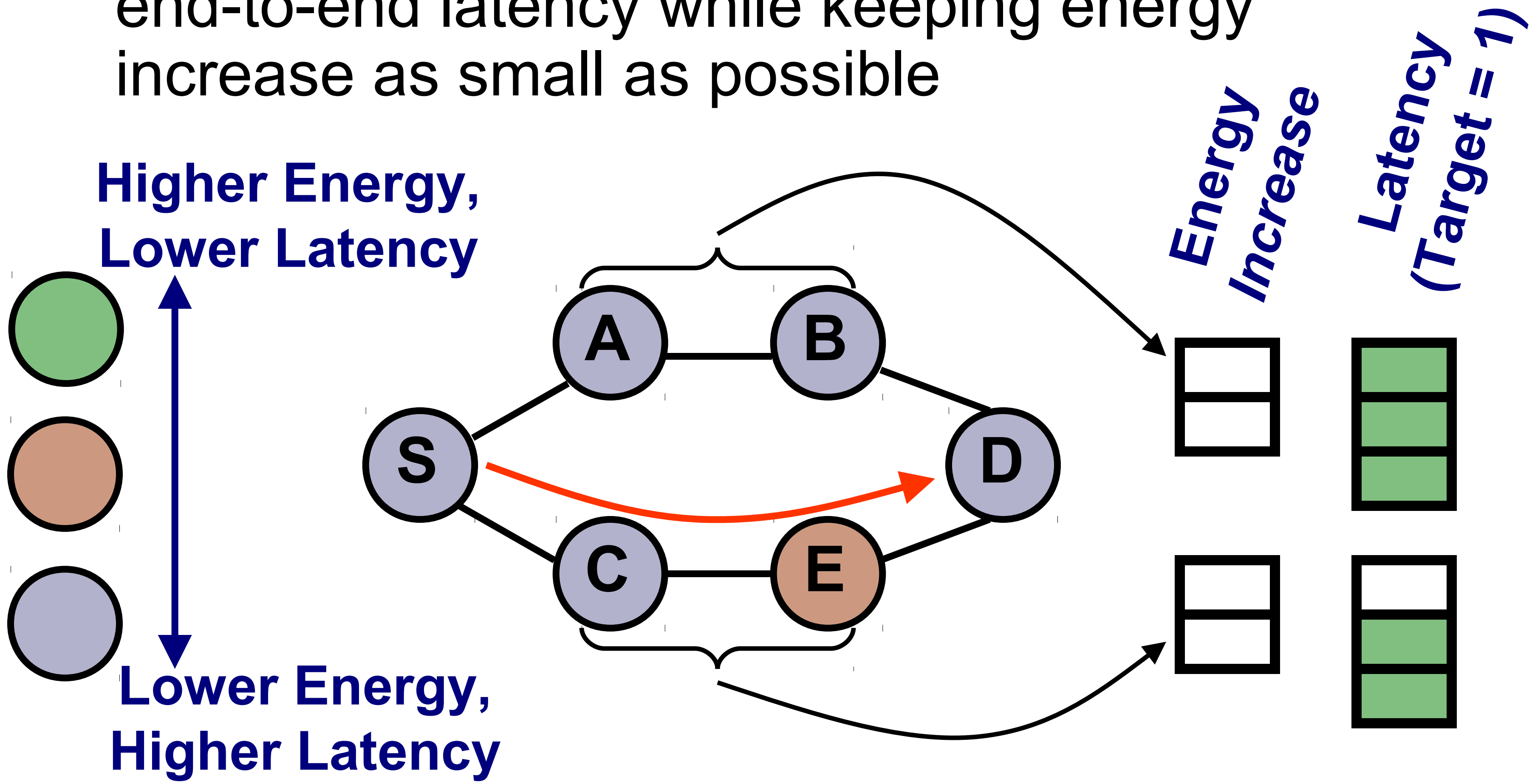
- A fixed listening interval can adversely affect energy efficiency, particularly as the load increases
- Adaptive listening significantly reduces energy consumption with only small increases in latency
- Carrier sense signaling is proposed and combined with adaptive listening to further improve energy efficiency

Thesis Outline



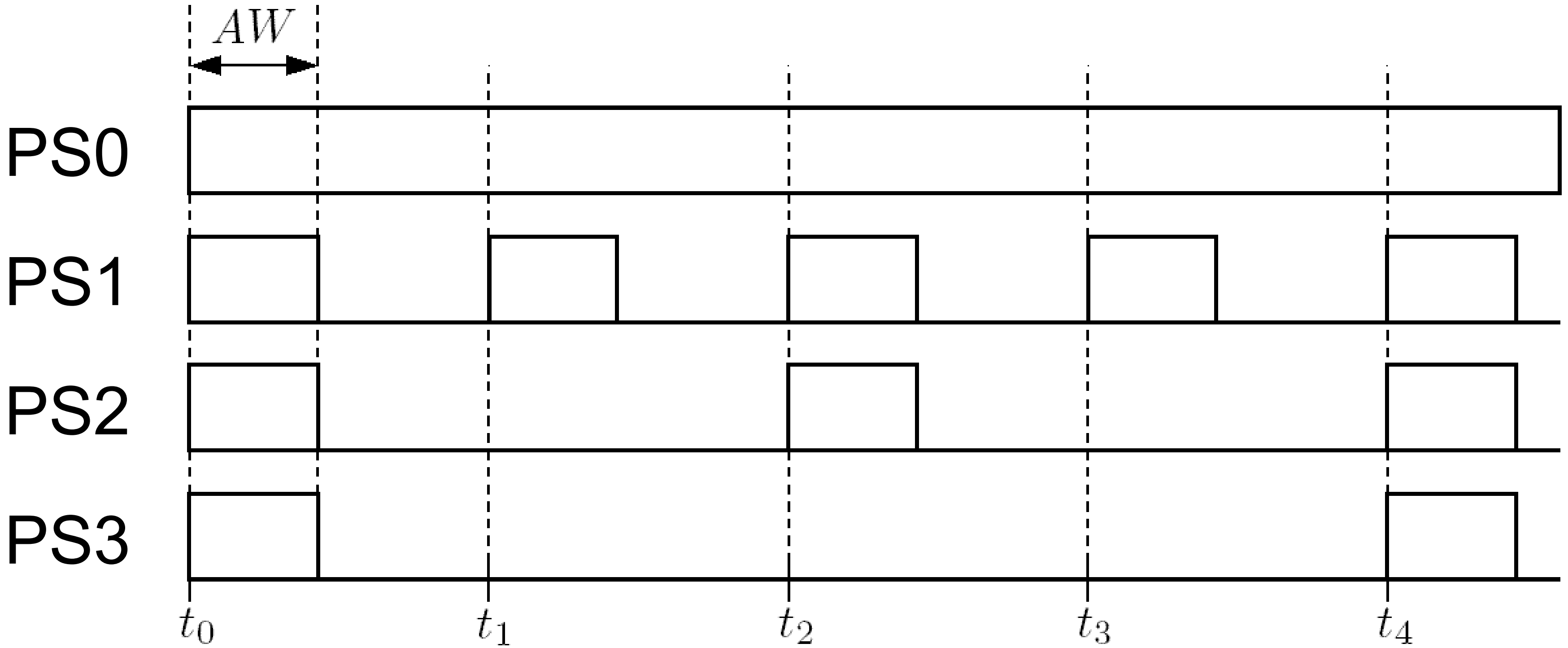
Adaptive Sleeping Overview

- Goal: Adapt sleeping interval to achieve desired end-to-end latency while keeping energy increase as small as possible



Multilevel Power Save (Link Layer)

Each power save state presents a different energy/latency tradeoff



Multilevel Power Save (Link Layer)

- Each level presents a different energy-latency tradeoff (i.e., higher energy → lower latency)
- 802.11 PSM
  - Nodes are synchronized to a reference point
  - T_S for *i*-th power level: $T_S(i) = 2^{i-1} * S_{base}$
    - $i > 0$ and $T_S(1) = S_{base}$
- Other PS protocols such S-MAC and WiseMAC can be modified similarly

Multilevel Power Save (Routing)

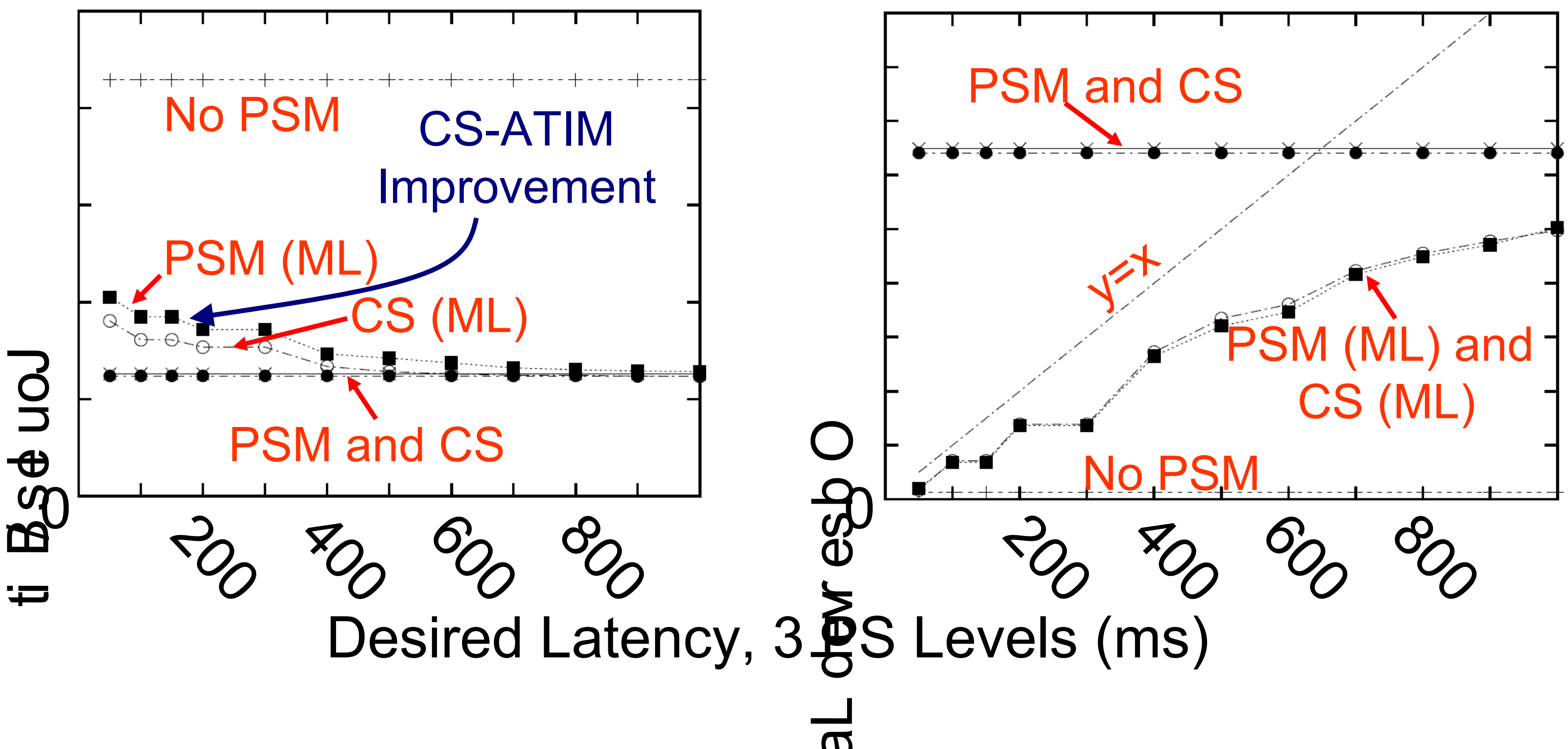
- We modify DSR to collect route requests for a specified duration
- For each collected path, iterate through the nodes
  - Find the minimum energy consumption increase required to achieve desired latency
- Select the path with the lowest required energy consumption increase

Adaptive Sleeping Results

- Simulated using *ns-2*
- Five flows with source and destination selected uniformly at random
  - Flow rate = 1 pkt/sec
  - S_{base} = 100 ms
- Routing protocol is DSR
- Link layer protocols are 802.11 PSM (PSM) and CS-ATIM (CS)
- All protocols tested with and without multilevel (ML) extension

Summary of Results

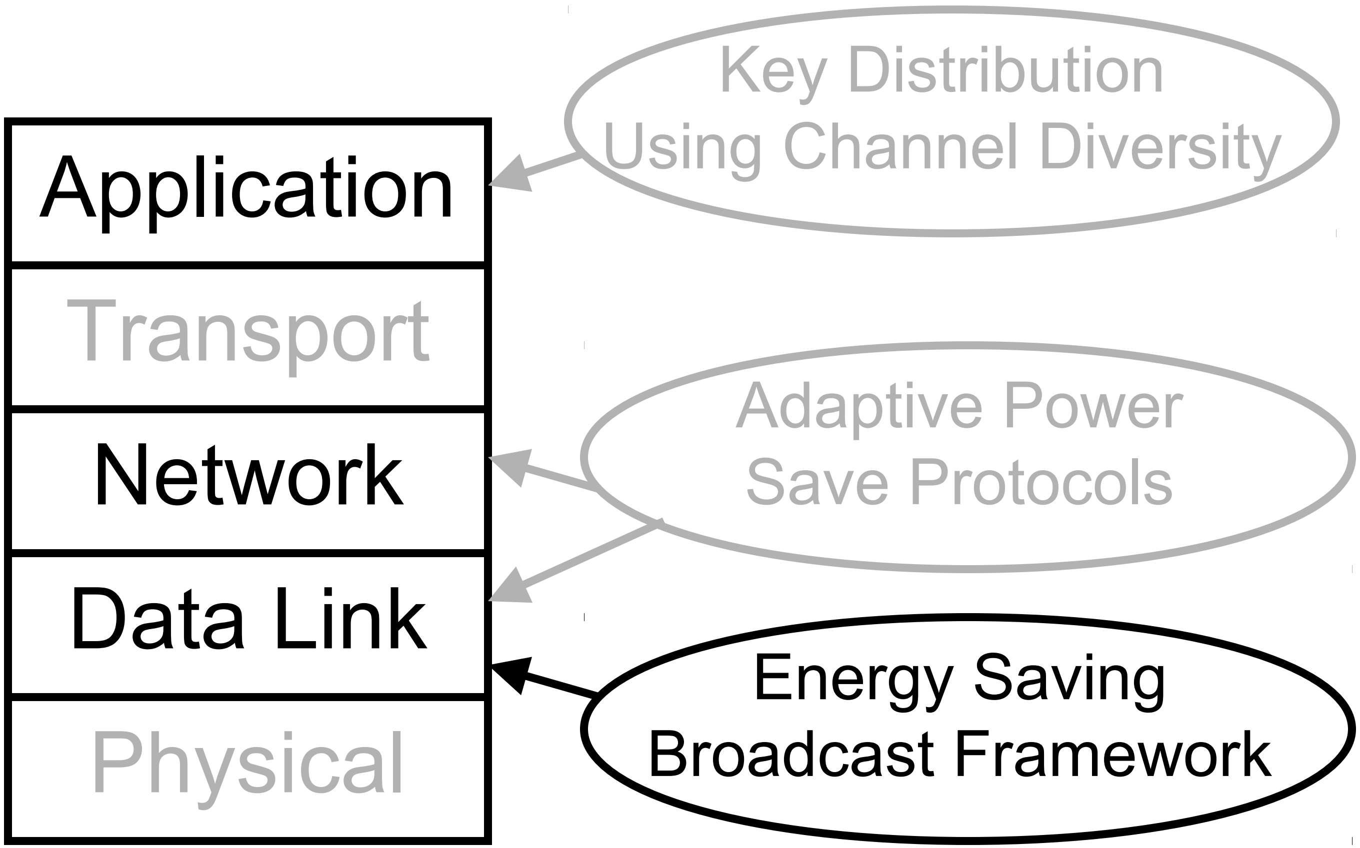
- ML maintains latency bound with only a small energy increase
- CS-ATIM further reduces energy with virtually no latency increase
- E.g., at 500 ms, CS-ATIM (ML) has the same energy consumption as the non-ML protocols with *half the latency*



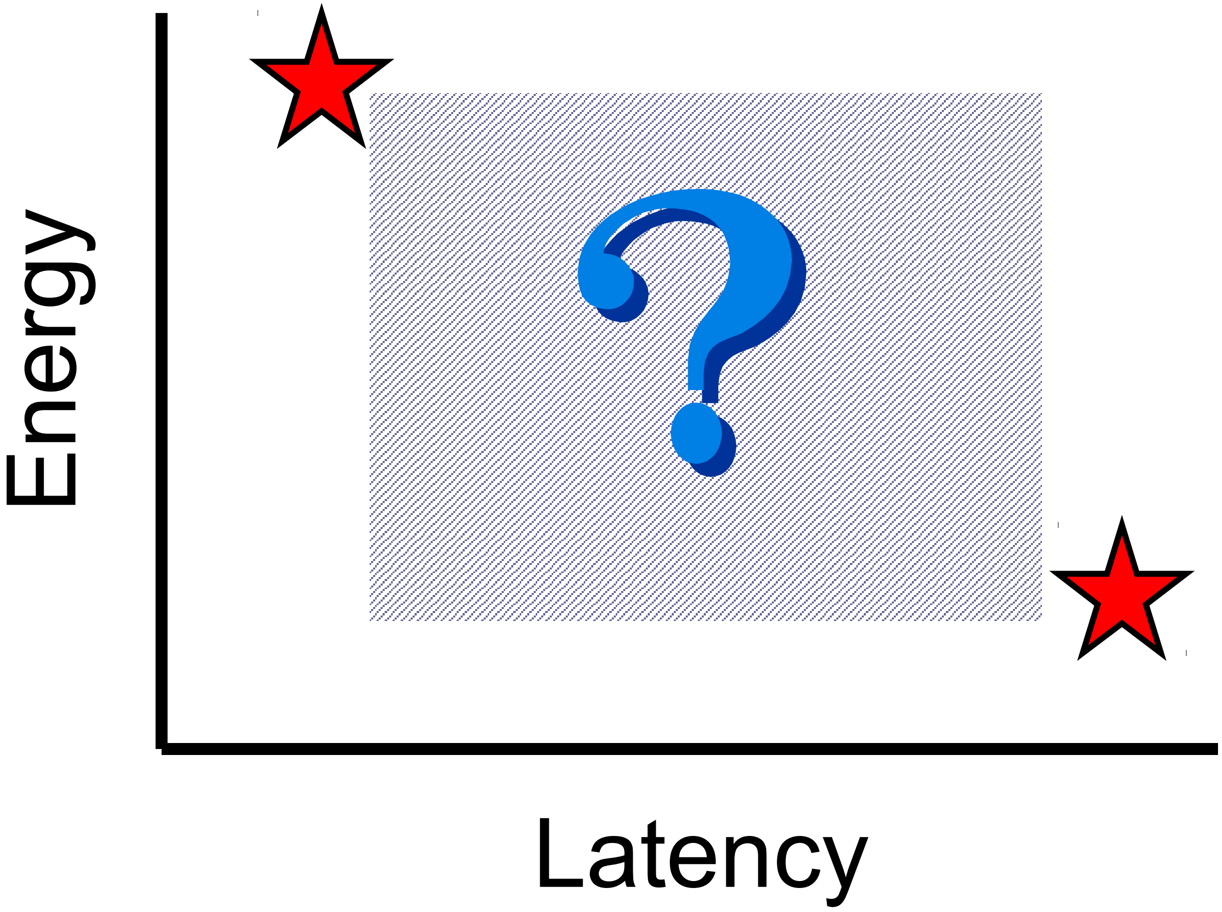
Summary

- Using a fixed sleeping interval can result in an unacceptable latency
- Adaptive sleeping can maintain an acceptable latency bound with relatively small degradations in energy consumption
- Our CS-ATIM protocol can further improve the energy efficiency with virtually no latency degradation

Thesis Outline



Multihop Broadcast: Energy-Latency Options



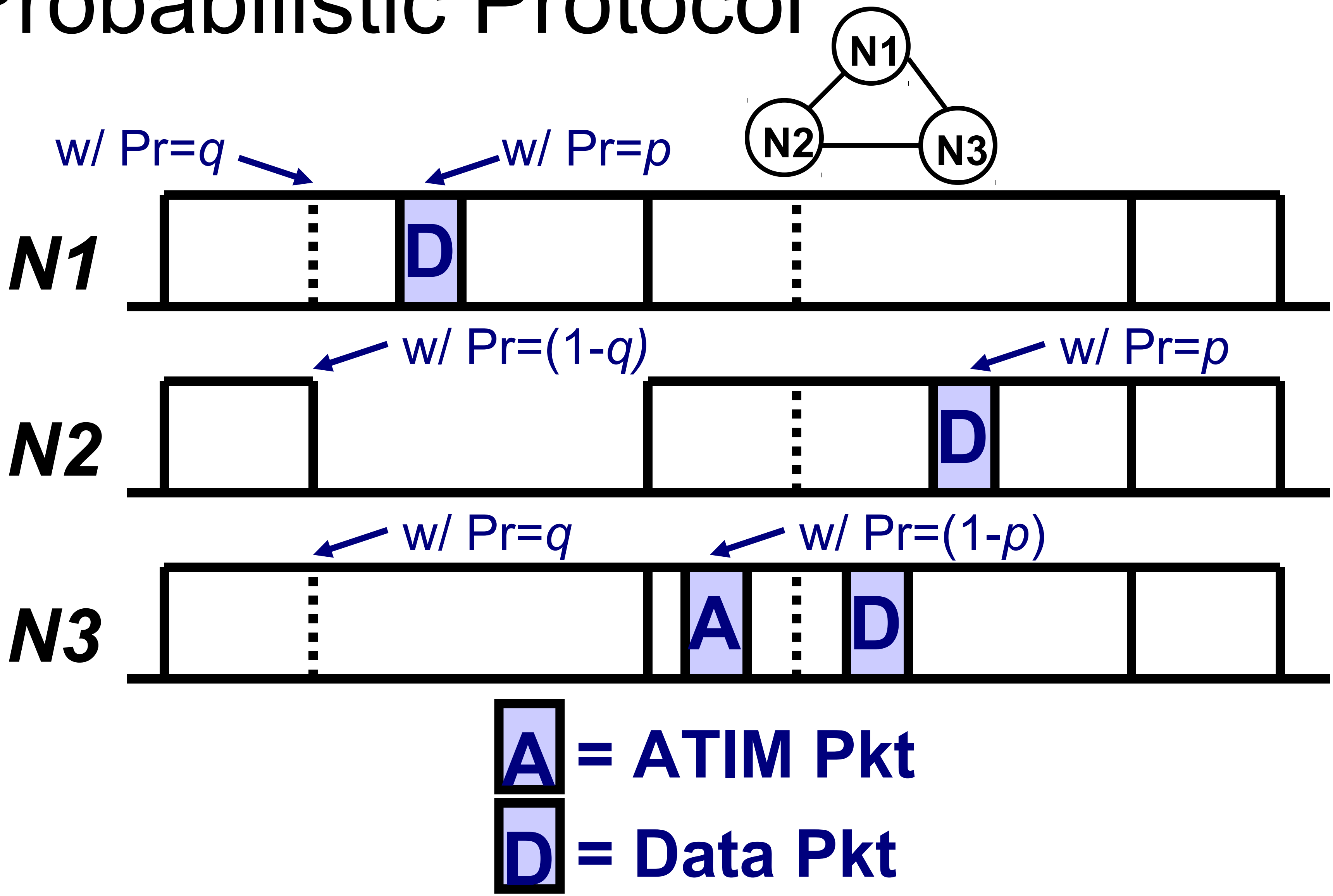
Our Work

- Design a protocol that allows users to adapt the energy-latency tradeoff to their needs for multihop broadcast applications
- Characterize the achievable latency and reliability performance for such applications that results from using power save protocols

Sleep Scheduling Protocols

- Nodes have two states: active and sleep
- At any given time, some nodes are active to communicate data while others sleep to conserve energy
- Examples
  - IEEE 802.11 Power Save Mode (PSM)
    - Most complete and supports broadcast
    - Not necessarily directly applicable to sensors
  - S-MAC/T-MAC
  - STEM

Probabilistic Protocol



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 BI before rebroadcasting the packet

Observations

- p=0, q=0 equivalent to the original sleep scheduling protocol
- p=1, q=1 approximates the “always on” protocol
  - Still have the ATIM window overhead

- Effects of p and q on metrics:

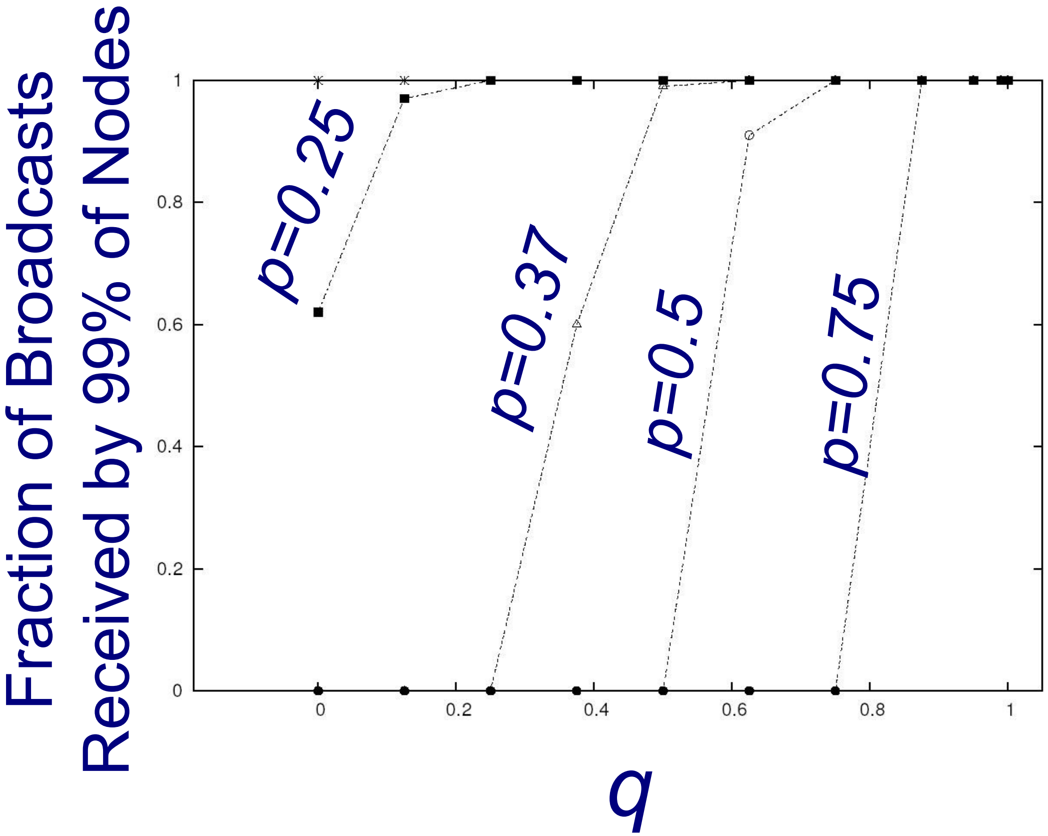
	Energy	Latency	Reliability
$p \uparrow$ (Immediate Send)	---	$\downarrow$ if $q > 0$	$\downarrow$ if $q < 1$
$q \uparrow$ (Stay On)	$\uparrow$	$\downarrow$ if $p > 0$	$\uparrow$ if $p > 0$

Summary of Results: Reliability

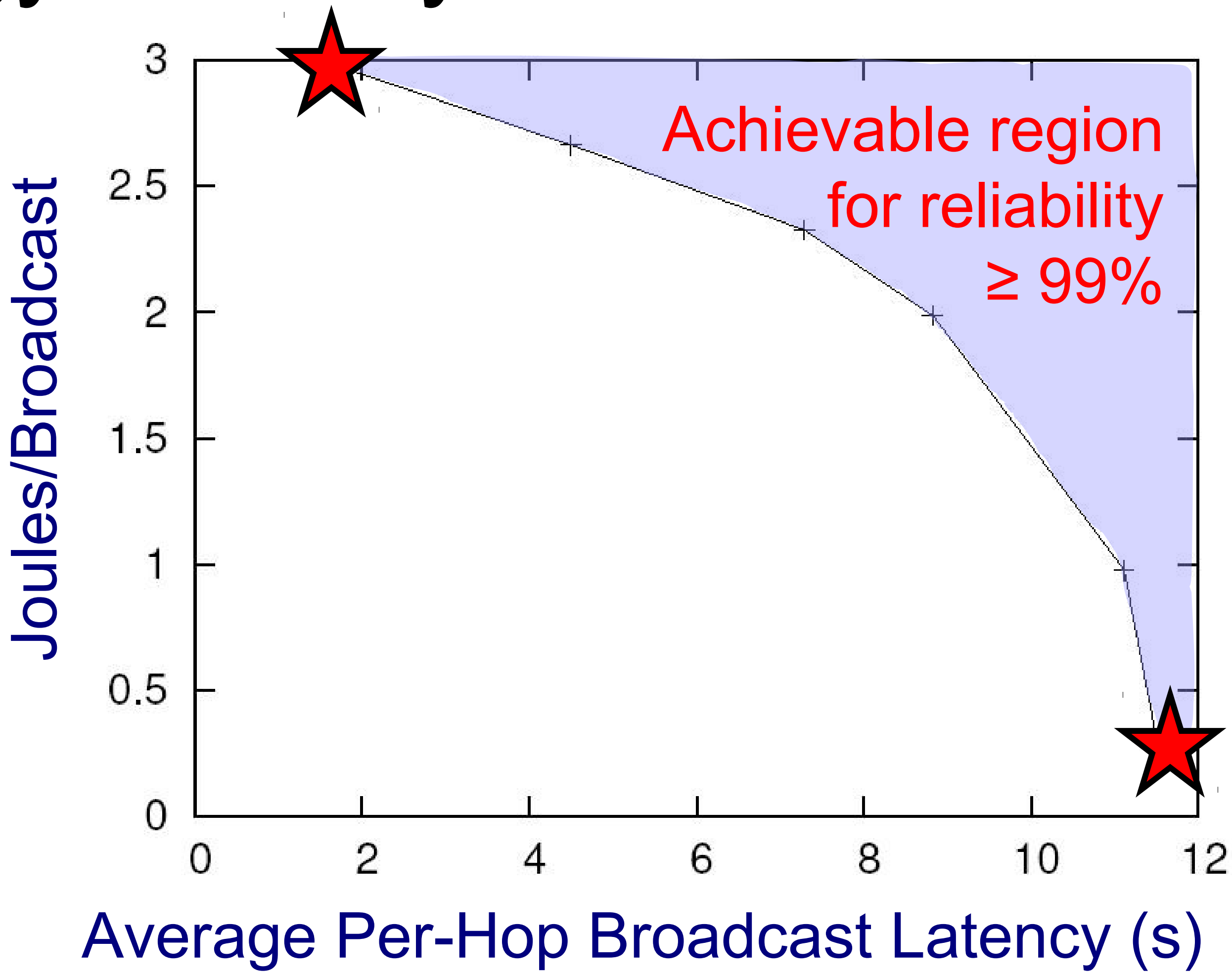
- Phase transition when:

$pq + (1-p) \approx 0.8\text{-}0.85$

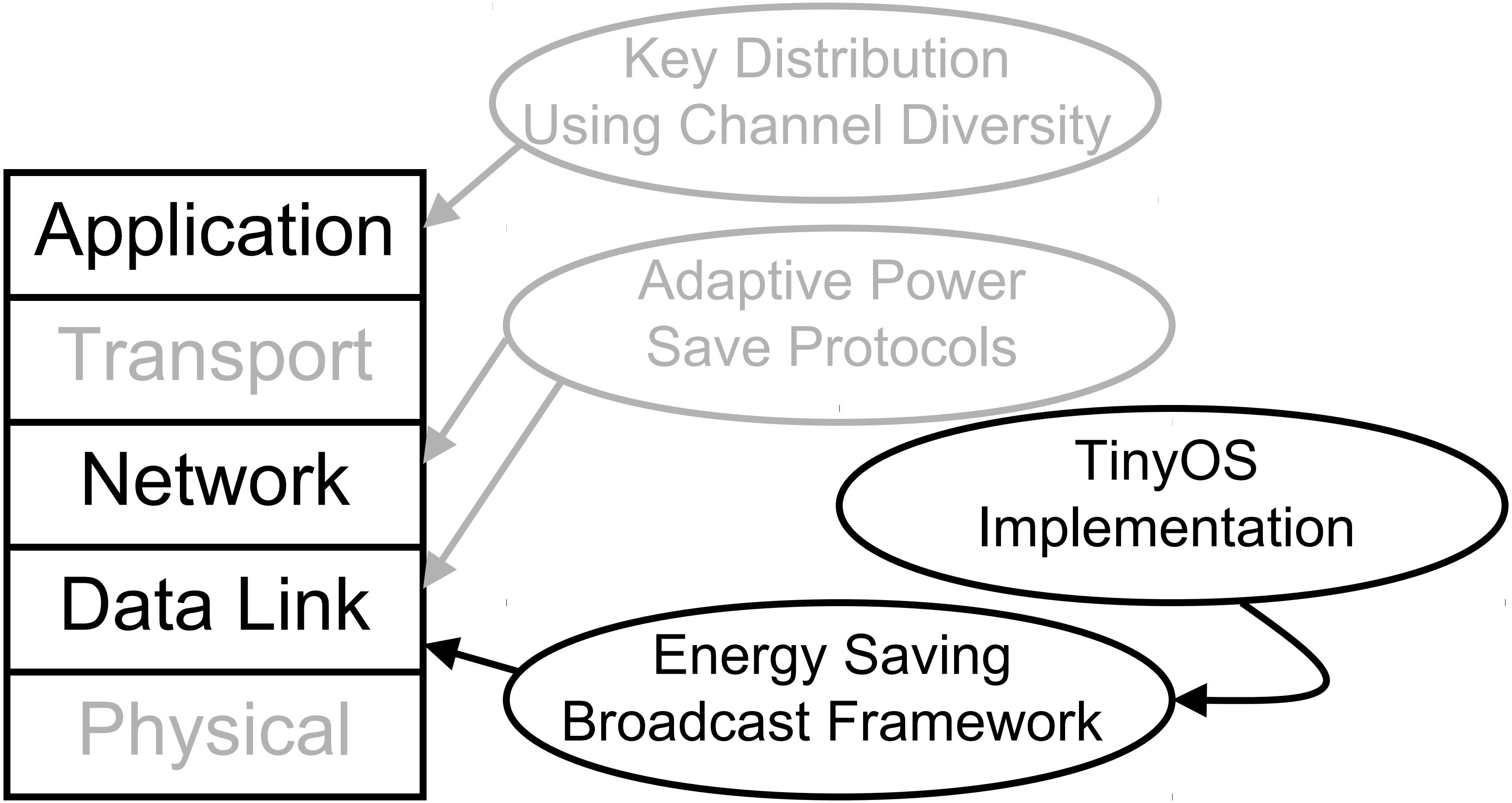
- Larger than bond percolation threshold (0.5)
  - Boundary effects
  - Different metric
- Still shows phase transition



Summary of Results: Energy-Latency Tradeoff

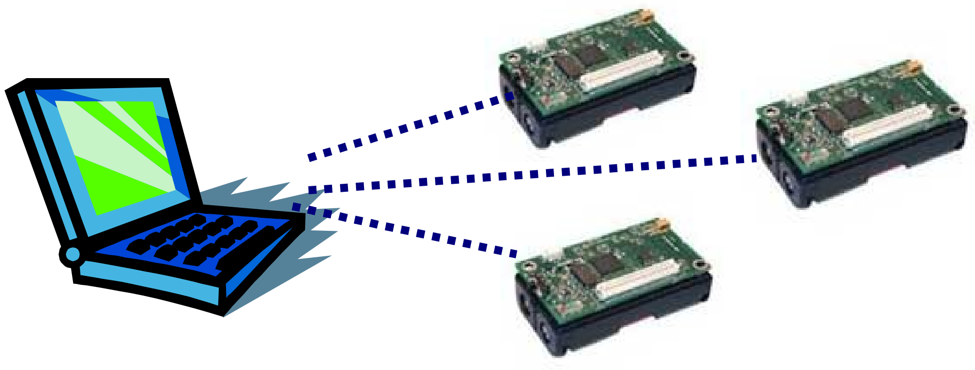


Thesis Outline

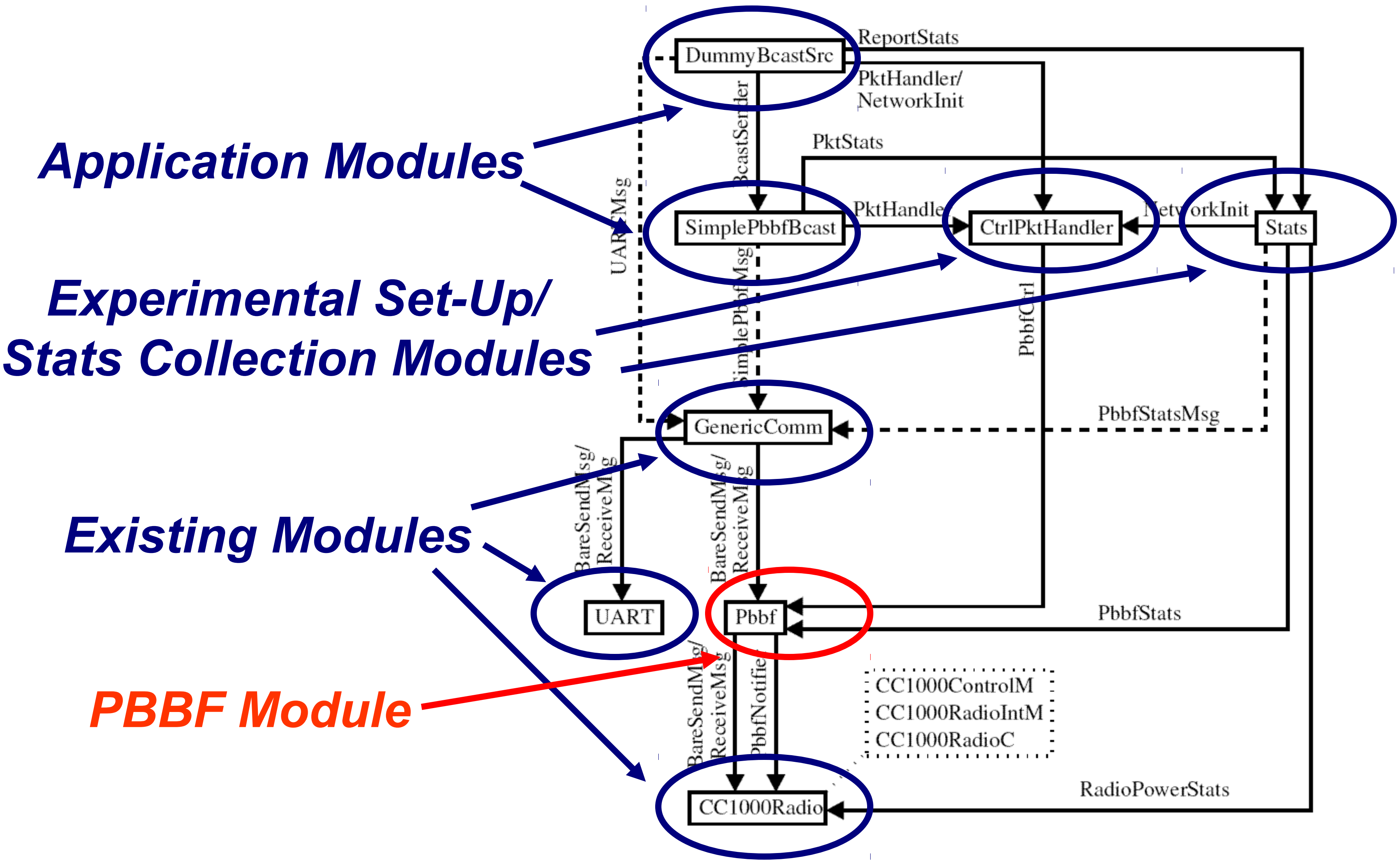


PBBF Implementation

- Used TinyOS on Mica2 Motes
  - Proof-of-concept
  - Application of PBBF to a different power save protocol (B-MAC)
  - Trends validate simulation results
- Extended PBBF by adding new parameter

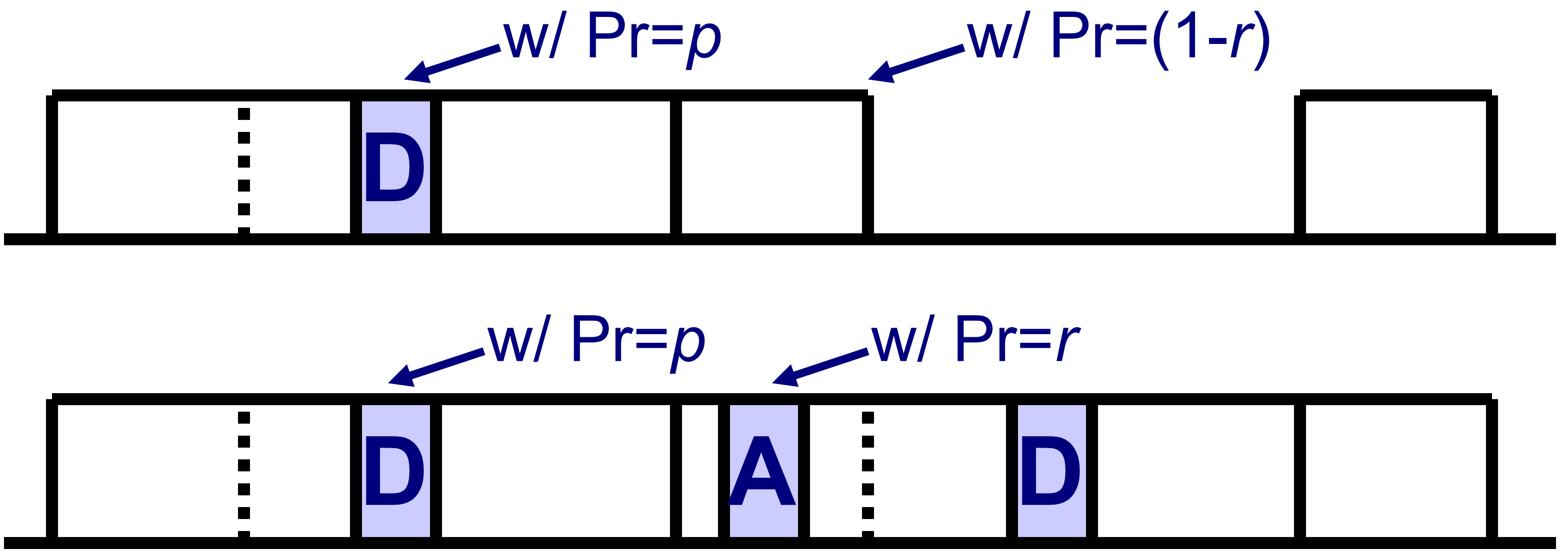


Our Architecture



PBBF Extension

- Added *r* parameter
  - If immediate send is done (with probability *p*), then, with probability *r*, retransmit the packet according to regular power save protocol
  - Tradeoff in reliability and overhead



Summary of Results

- Confirm trends in simulation and analysis
- The *r* parameter improves reliability, but increases energy consumption, latency, and overhead

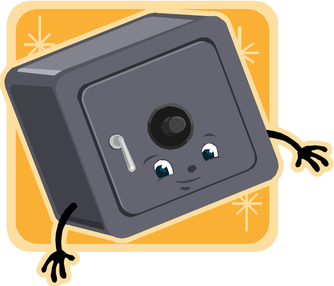
	Energy	Latency	Reliability	Overhead
p ↑ if $r = 0$	---	↓ if $q > 0$	↓ if $q < 1$	---
q ↑ if $r = 0$	↑	↓ if $p > 0$	↑ if $p > 0$	---
r ↑ if $p > 0$	↑	↑	↑	↑

Summary

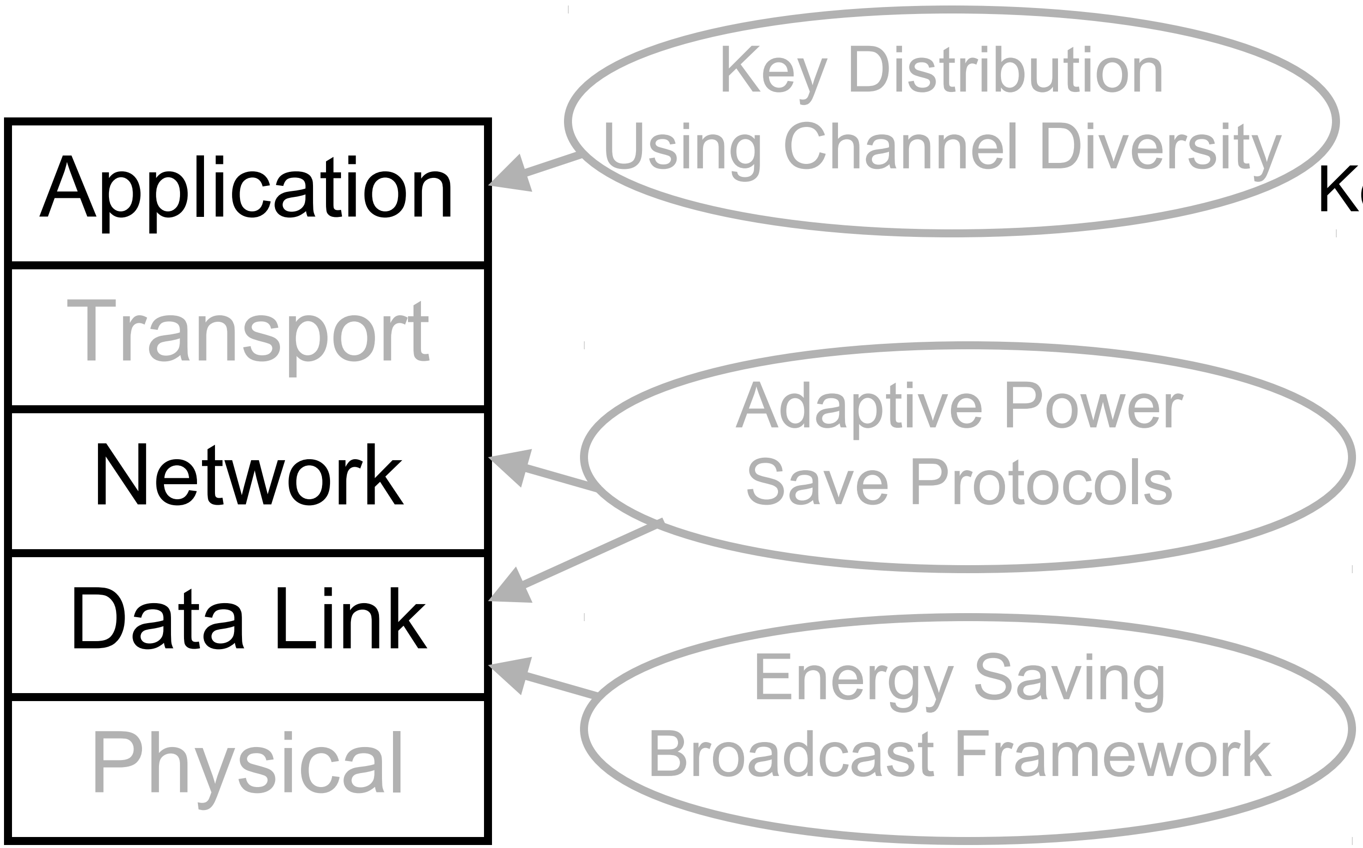
- Shown the effects of energy-saving on the latency and reliability of applications that disseminate data via multihop broadcast
- Designed protocol that allows wide range of tradeoffs for such applications
- Implemented protocol in TinyOS and quantified performance
- ***Acknowledgements****: Joint work done with Cigdem Sengul and Indranil Gupta*

Thesis Goals: Energy Efficiency and Security

- Both areas need significant improvement for ubiquitous wireless networks to become a reality

- Energy Efficiency: Marginal gains in batteries necessitate power save protocols
- Security: Resource constrained devices with insecure wireless channels


Thesis Outline



Key Distribution Background

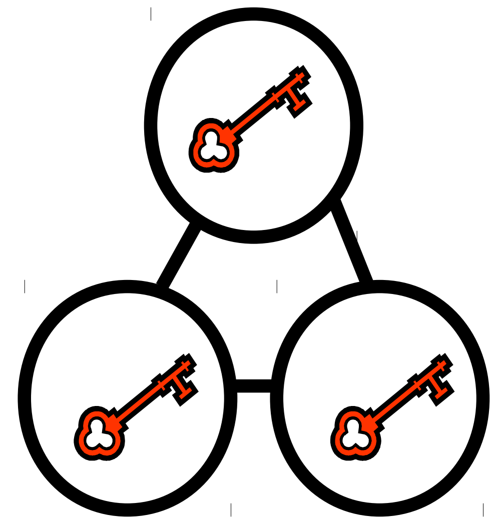
Problem Statement

- After deployment, a sensor needs to establish pairwise symmetric keys with neighbors for confidential and authenticated communication

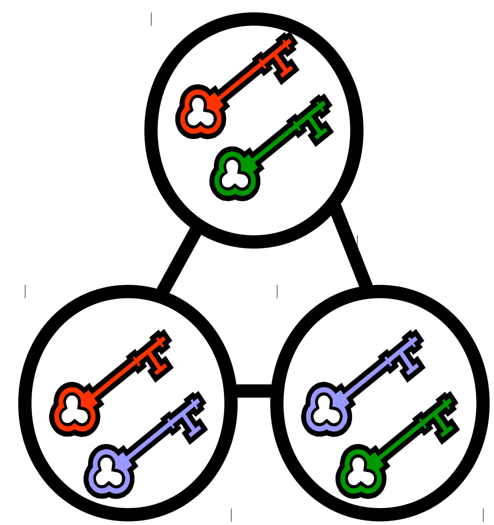
- Applications
  - Secure aggregation
  - Exchanging hash chain commitments (e.g., for authenticated broadcast)

Design Space

- Every node deployed with global key
  - ☺ Minimal memory usage, incremental deployment is trivial
  - ☹ If one node is compromised, then all links are compromised



- Separate key for each node pair
  - ☺ One compromised node does not affect the security of any other links
  - ☹ Required node storage scales linearly with network size



Related Work

- Each sensor shares a secret key with a trusted device (*T*) [Perrig02Winet]
  - *T* used as intermediary for key establishment
  - *T* must be online and may become bottleneck

- Key Predistribution [Eschenauer02CCS]
  - Sensors pre-loaded with subset of keys from a global key pool
  - Tradeoff in connectivity and resilience to node compromise
  - Each node compromise reduces security of the global key pool

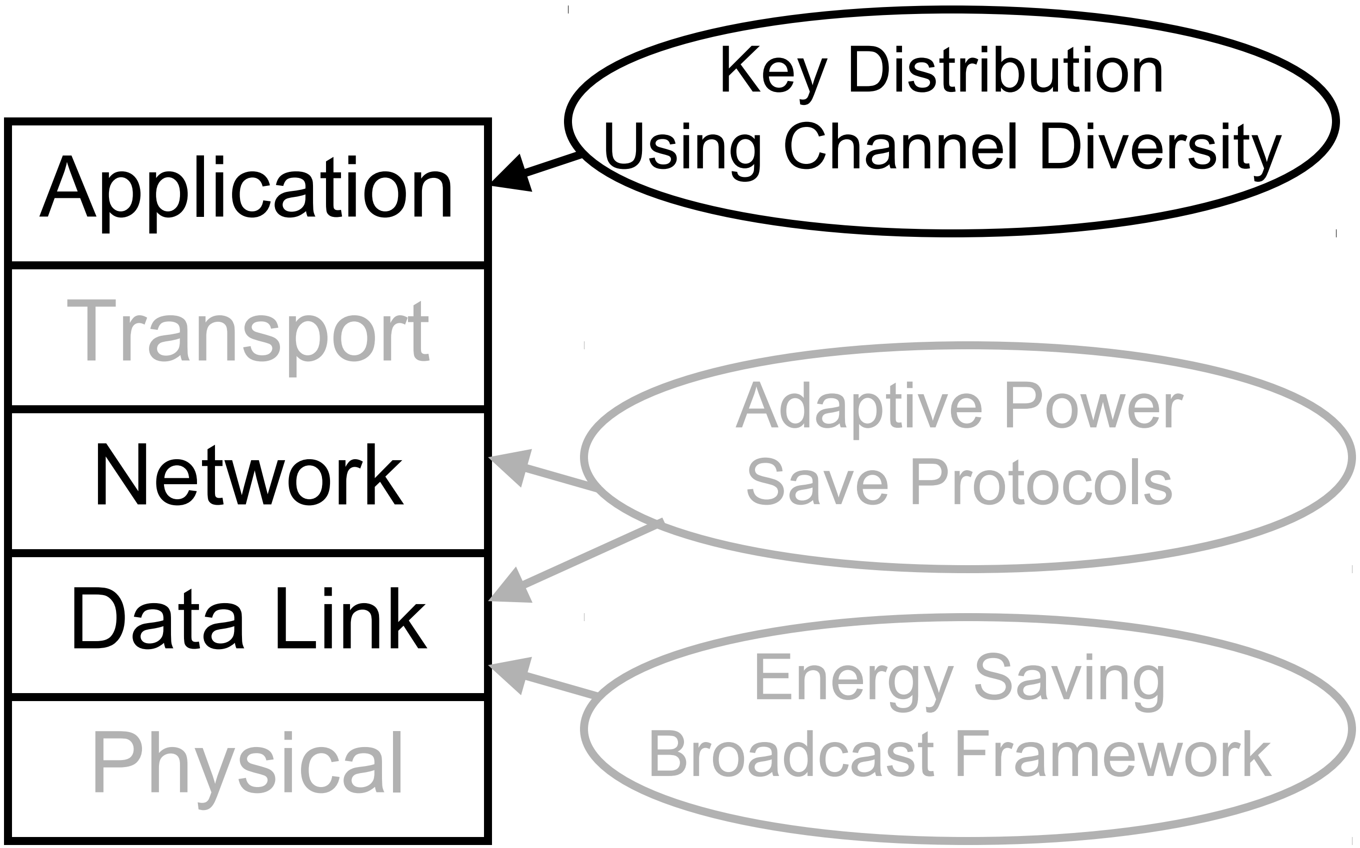
Related Work

- Transitory key [Zhu03CCS]
  - Sensors use global key to establish pairwise key and then delete global key
  - Node compromise prior to deletion could compromise entire network
- Using public keys (e.g., Diffie-Hellman)
  - High computation cost
  - But, is it worth it when this cost is amortized over the lifetime of a long-lived sensor network?

Related Work

- Broadcast plaintext keys [Anderson04ICNP]
  - If an eavesdropper is not within range of both communicating sensors, then the key is secure
  - Assumes very small number of eavesdroppers
  - No way to improve link security if eavesdroppers are in range
  - We propose using the underlying wireless channel diversity to greatly improve this solution domain

Thesis Outline



High Level View of Our Work

High Level View of Our Work

- Given c channels:
  Pr(Eve hears Bob's packet | Alice hears Bob's packet) = $1/c$
- If Alice hears M of Bob's packets, then the probability that Eve heard *all* of those packets is $(1/c)^M$
- As $(1/c)^M \to 0$:
  The packets Alice heard can be combined to create Alice and Bob's secret key

Threat Model

- Adversary's primary objective is to learn pairwise keys
  - Can compromise node and learn its known keys
  - Can overhear broadcast keys
- Adversary's radio capability is similar to that of sensors [Anderson04ICNP]
  - Receive sensitivity
  - One radio
- Multiple adversary devices may collude in their knowledge of overheard keys
  - Collusion in coordination of channel listening is future work
- Denial-of-Service is beyond the scope of our work

Protocol Overview

- Predeployment
  - Give each sensor a unique set of authenticatable keys
- Initialization
  - Broadcast keys to neighbors using channel diversity
- Key Discovery
  - Find a common set of keys shared with a neighbor
- Key Establishment
  - Use this set to make a pairwise key that is secret with high probability

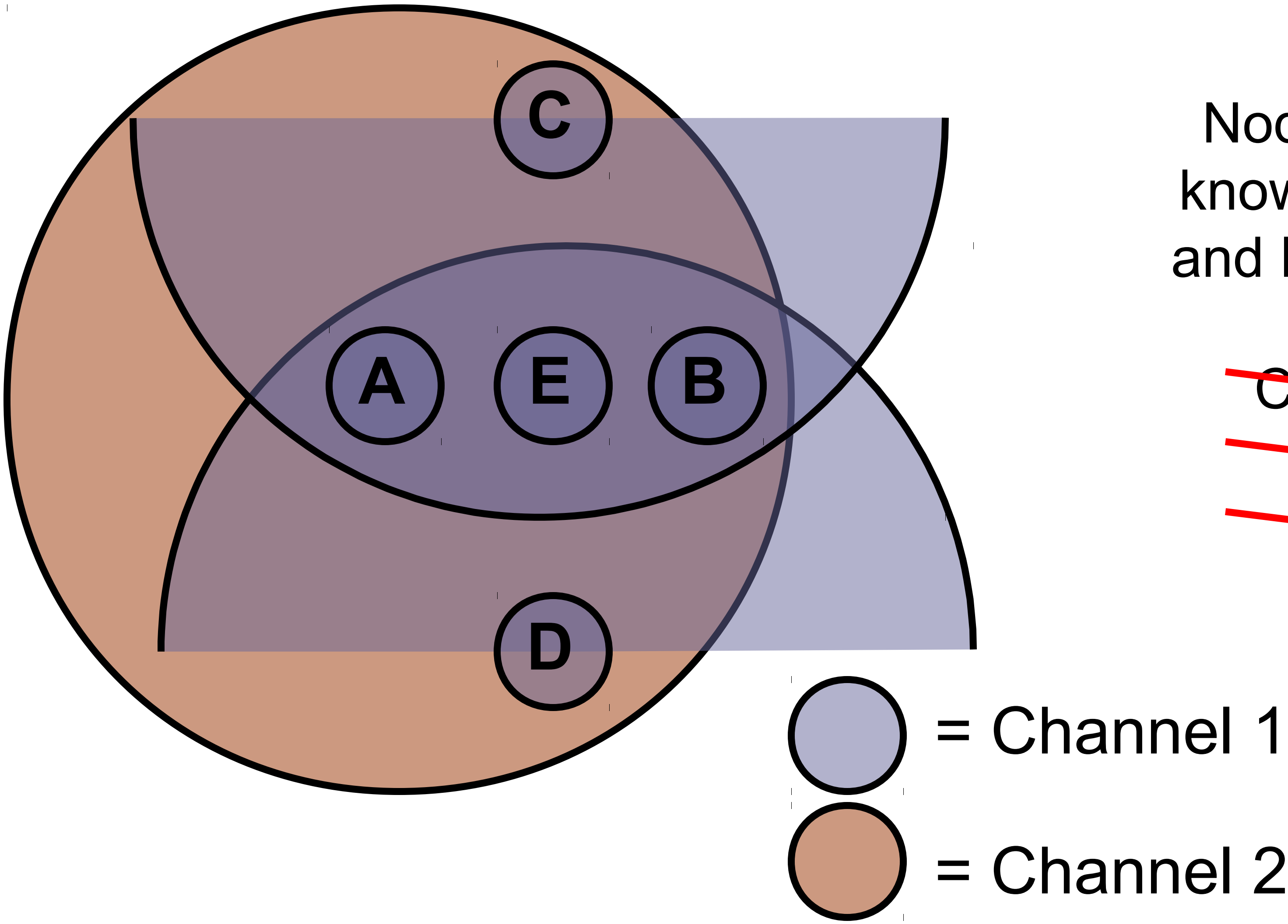
Phase 1: Predeployment

- Each sensor is given λ keys by a trusted entity
  - Keys are unique to sensor and *not* part of global pool
  - λ presents a tradeoff between overhead and security
- The trusted entity also loads the Merkle tree hashes needed to authenticate a sensor's keys
  - O(lg *N*) hashes using Bloom filter authentication
  - O(lg λ*N*) hashes using direct key authentication

Phase 2: Initialization

- Each sensor follows two unique non-deterministic schedules:
  - When to switch channels
    - Chosen uniformly at random among *c* channels
  - When to broadcast each of its λ keys
- Thus, each of a sensor's λ keys is overheard by *1/c* neighbors on average
  - Different subsets of neighbors overhear each key
- Sensors store every overheard key

Initialization Example



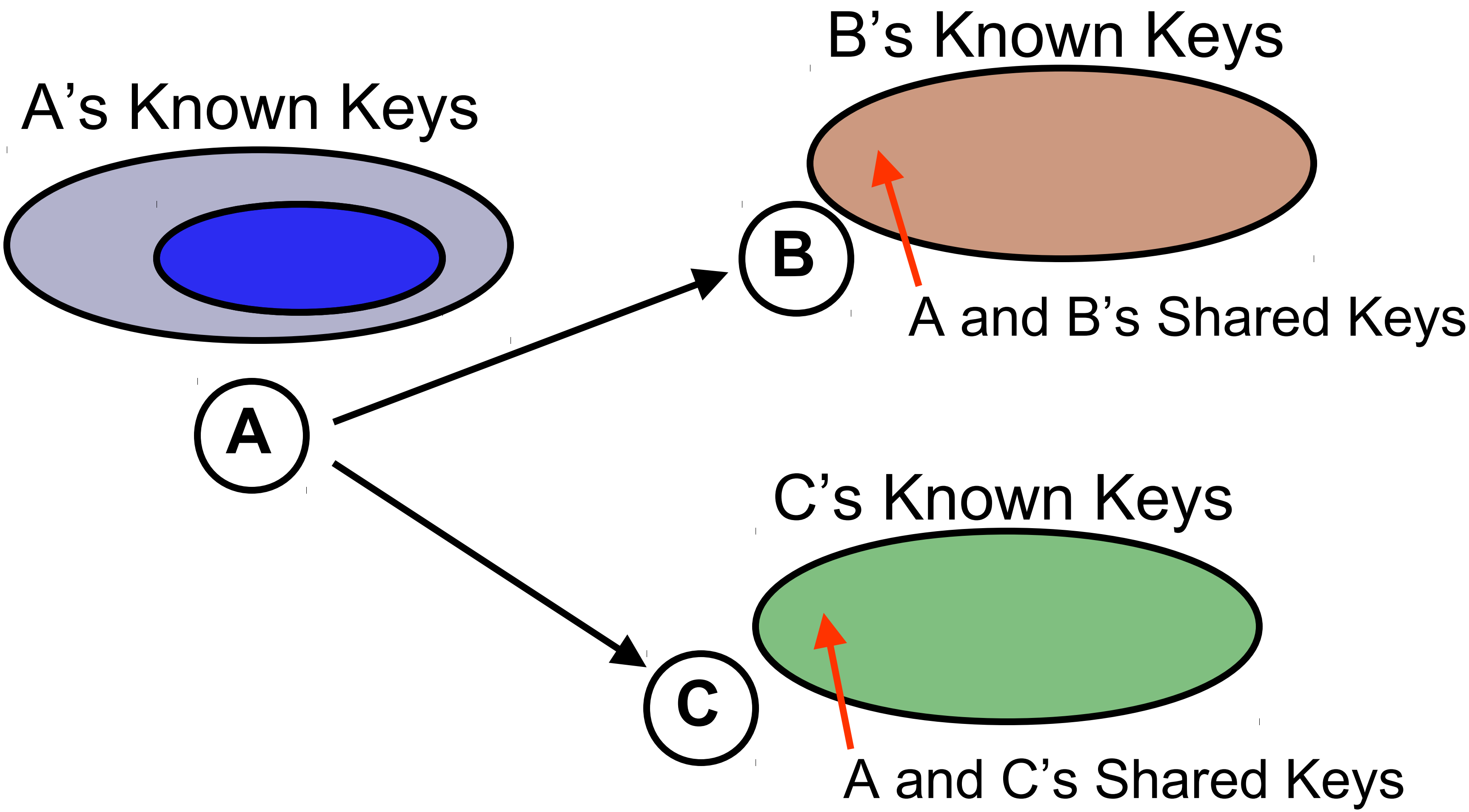
Nodes that know all of A and B's keys:

~~C, D, E~~
~~C, E~~
~~E~~
Ø

Phase 3: Key Discovery

- Goal: Discover a subset of stored keys known to each neighbor
- All sensors switch to common channel and broadcast Bloom filter with β of their stored keys
  - Bloom filter for reduced communication overhead
- Sensors keep track of the subset of keys that they believe they share with each neighbor
  - May be wrong due to Bloom filter false positives

Key Discovery Example

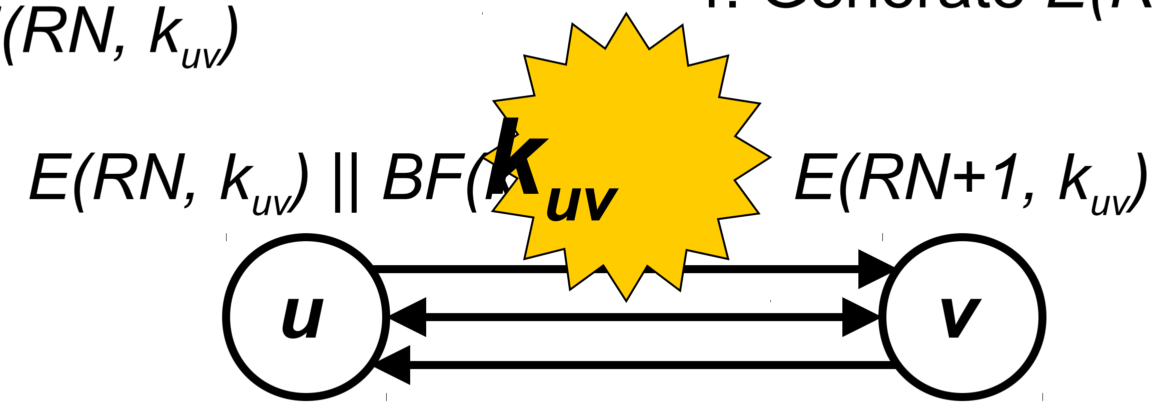


Phase 4: Key Establishment

u's believed set of shared keys with *v* = $\{k_1, k_2, k_3\}$

1. Generate link key: $k_{uv} = hash(k_1 \| k_2 \| k_3)$
2. Generate Bloom filter for k_{uv}: $BF(k_{uv})$
3. Encrypt random nonce (*RN*) with k_{uv}: $E(RN, k_{uv})$

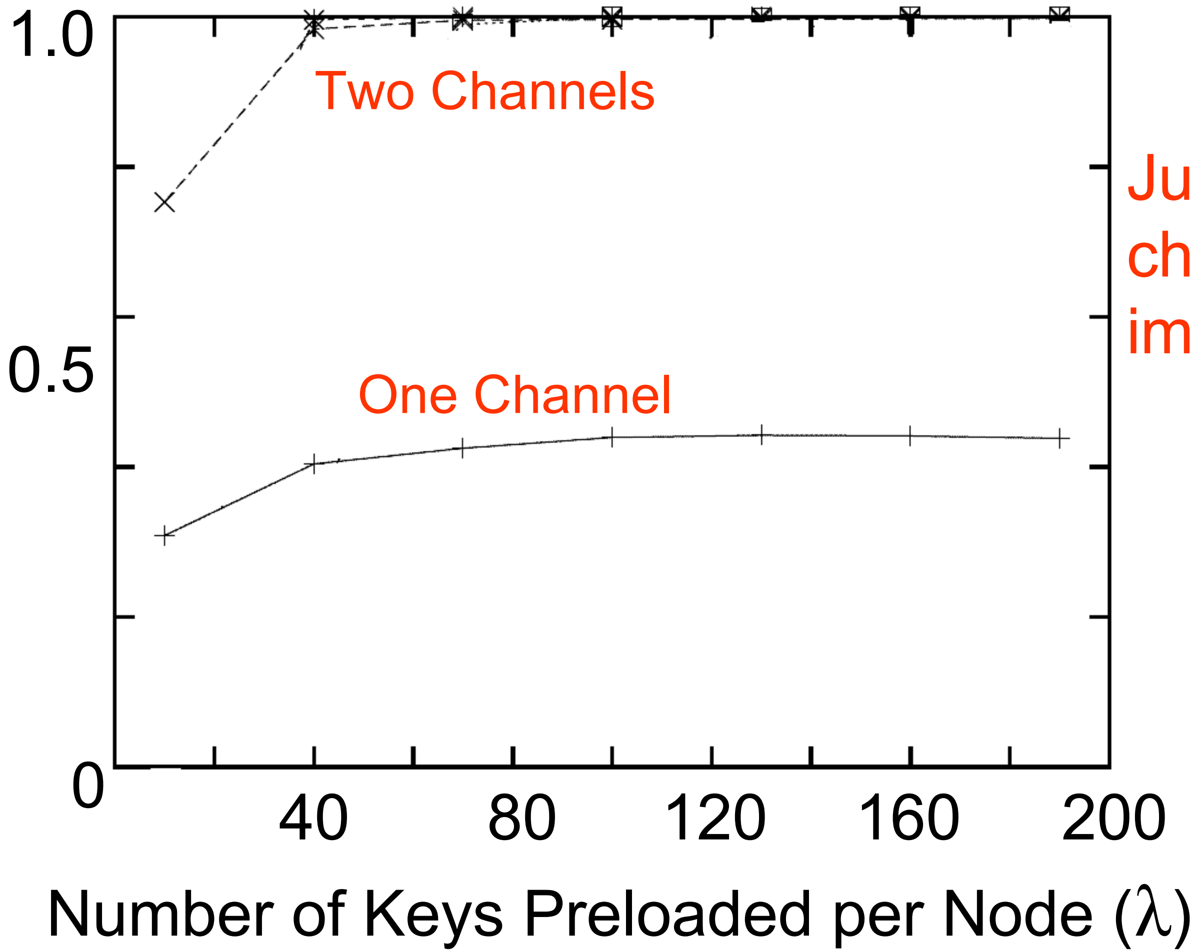
1. Find keys in $BF(k_{uv})$
2. Use keys from Step 1 to generate k_{uv}
3. Decrypt $E(RN, k_{uv})$
4. Generate $E(RN+1, k_{uv})$



Simulation Setup

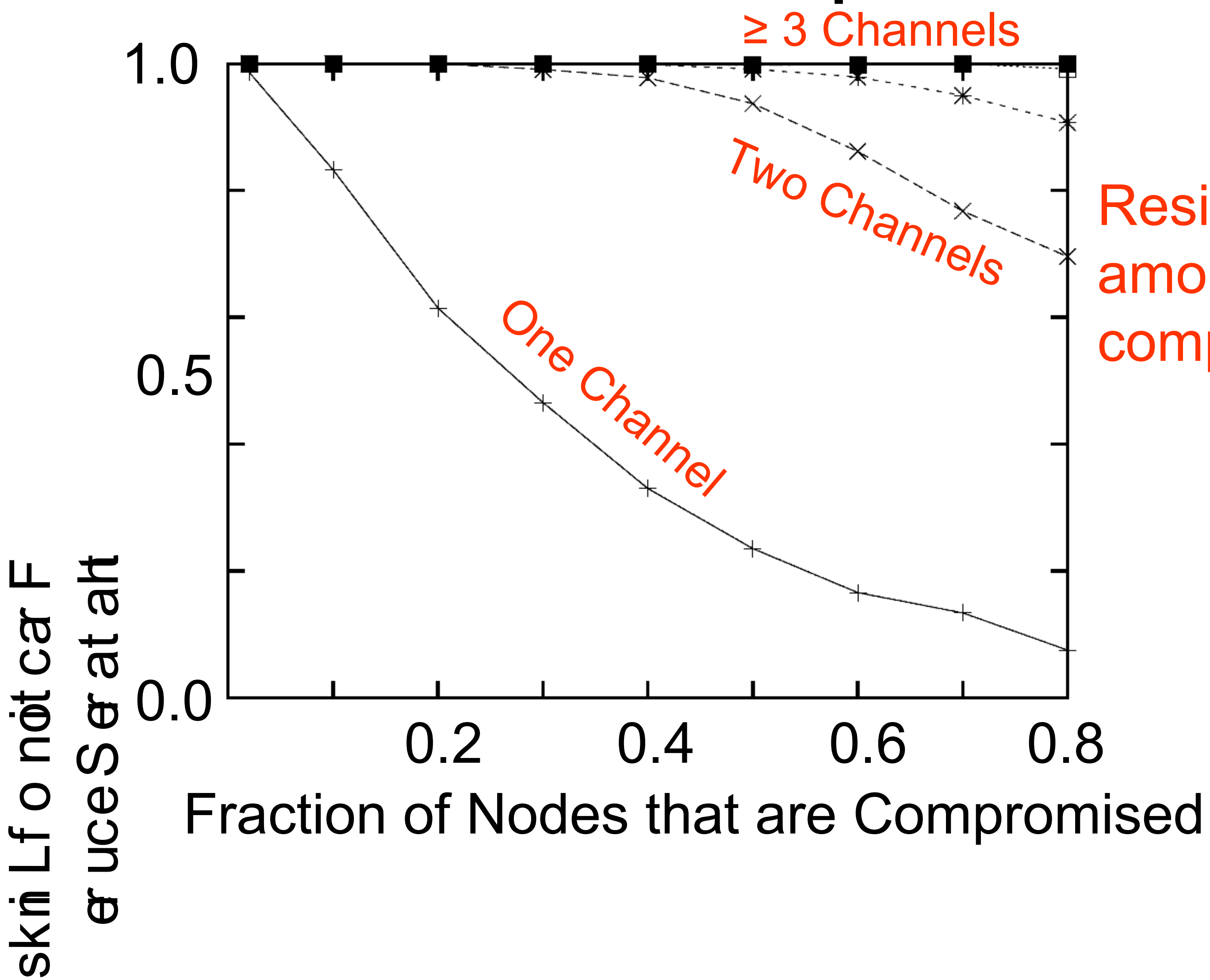
- Use *ns-2* simulator
- 50 nodes
- Density of 10 expected one hop neighbors
- By default, 15 nodes are adversaries and collude in their key knowledge
- By default, λ is 100 keys/sensor

Summary of Results: The Advantage of Channel Diversity



Just one extra channel significantly improves security

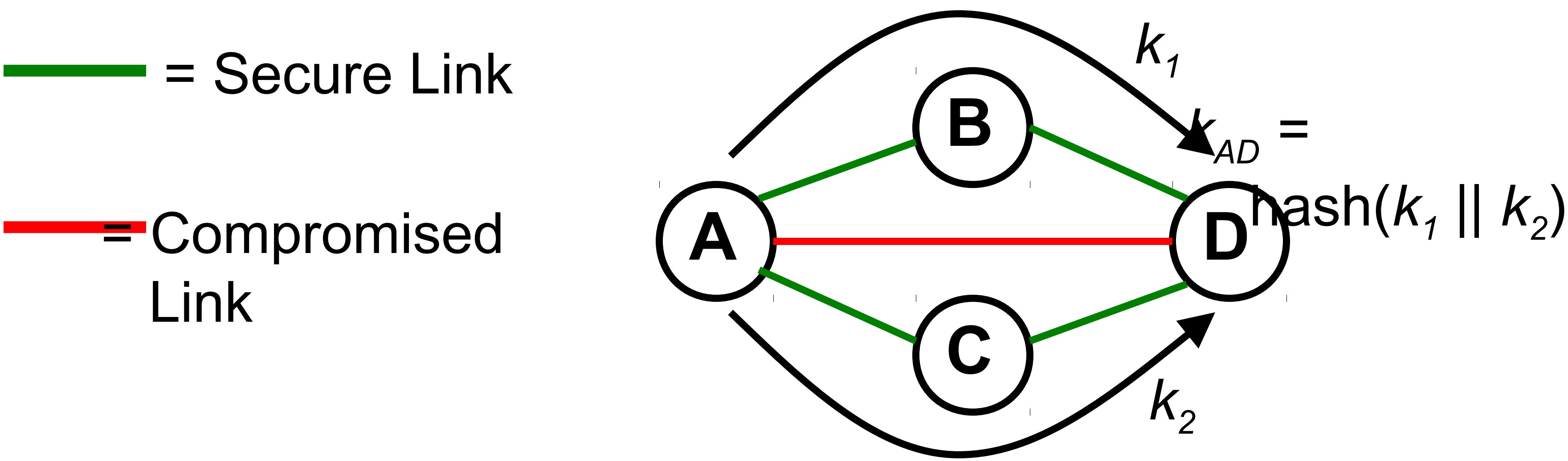
Summary of Results: Resilience to Compromise



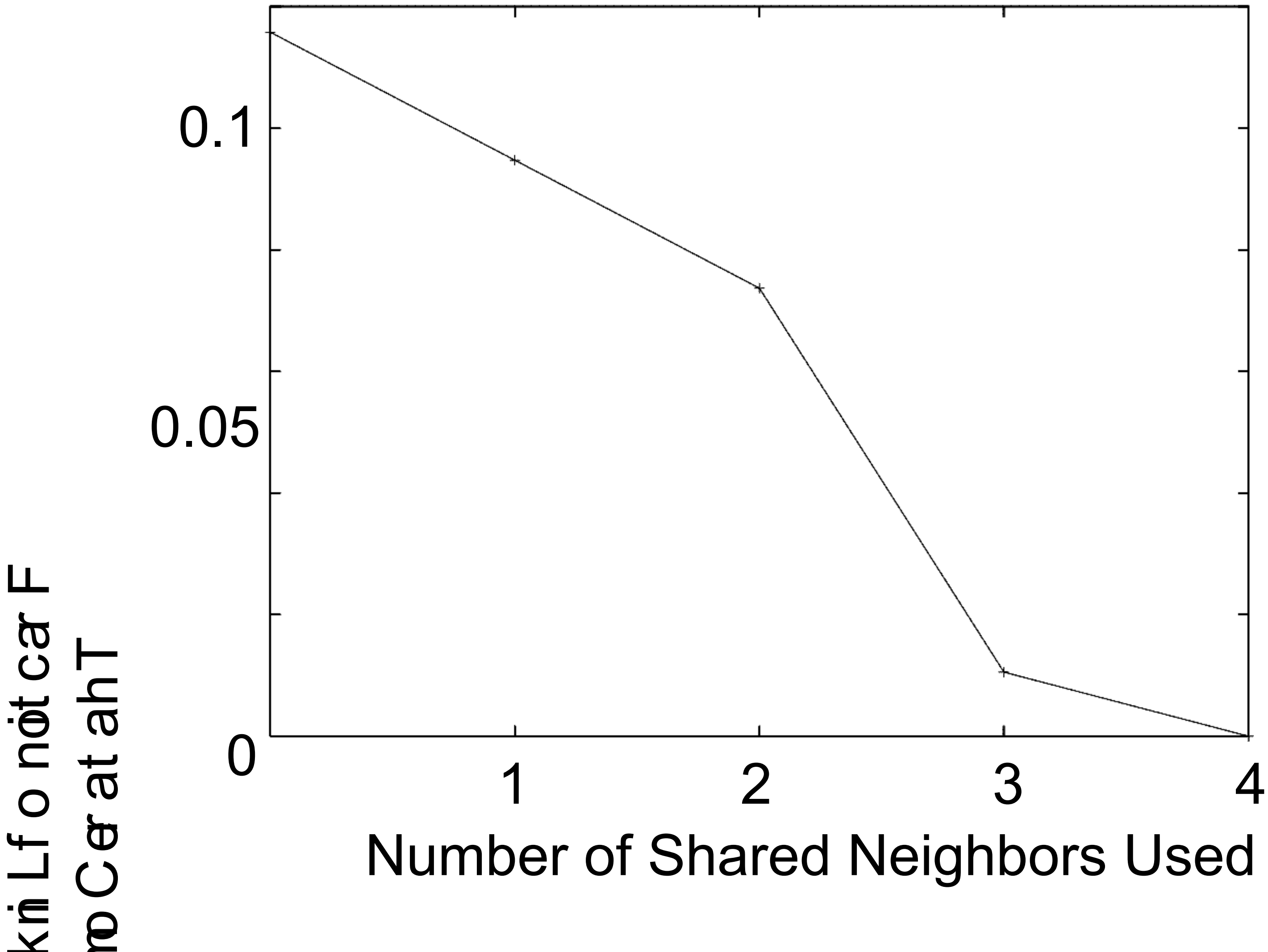
Resilient to large amount of node compromise

Using Path Diversity

- Path diversity can be used to get a small number of compromised links to zero
- Similar to multipath reinforcement proposed elsewhere
  - Node disjoint paths needed to combat node compromise
  - Only link disjoint paths needed to combat eavesdroppers



Simulation Results for Example Topology



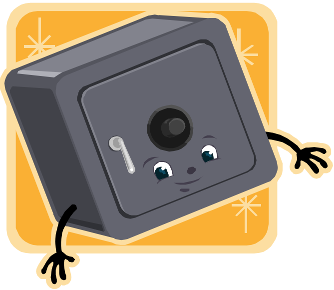
Summary

- Many distinct solutions have been proposed
  - No “one size fits all” approach emerges
- Our work is the first to propose using channel diversity for key distribution
  - Results show significant security gains when even *one* extra channel is used
- Path diversity can further improve key security

Thesis Conclusion

Energy efficiency and security are major issues facing multihop wireless networks

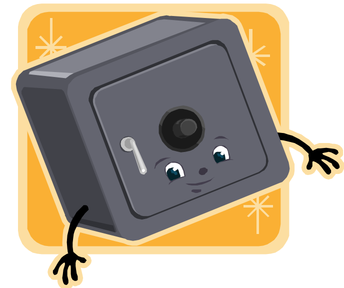
- Energy Efficiency
  - Battery energy-density has shown little improvement
  - The radio is a major power sink in small/no display devices
- Security
  - Smaller devices are resource constrained
  - Node compromise is relatively easy



Thesis Conclusion: Energy Efficiency

- Carrier sensing is effective at reducing energy consumption for wake-up signaling
  - Proposed for both in-band and out-of-band protocols
- Adaptive listening and sleeping protocols dynamically modify parameters in response to the current environment
  - Offers improvements over fixed parameter protocols
- Broadcast framework allows fine-grained control over energy, latency, and reliability
  - Tradeoffs quantified via simulation and implementation

Thesis Conclusion: Security



- Key distribution in sensor networks provides confidentiality and authentication
  - Resource constraints favor symmetric key operations which makes distribution difficult
- We are the first to propose leveraging channel diversity for this task
  - Results show both good connectivity *and* resilience to node compromise when compared to previous work

Open Research Problems

- Energy Efficiency
  - Implementing our power save protocols and testing them in the context of an application-layer task
  - Designing power save for multichannel and multi-interface protocols
- Security
  - Analyzing quantitative tradeoffs of pure symmetric key exchange versus public key exchange
  - Exploring other techniques that use wireless diversity for security

Thank You!

http://www.crhc.uiuc.edu/~mjmille2

mjmille2@uiuc.edu

Sources (Ordered by First Appearance)

- *The Other Wireless Revolution* by David A. Gross
  - http://www.state.gov/e/eb/rls/rm/2005/48757.htm
- *Report: RFID production to increase 25 fold by 2010* in EE Times
  - http://tinyurl.com/aangg
- *CNET's quick guide to Bluetooth headsets* on CNET.com
  - http://tinyurl.com/dslev
- TinyOS Community Forum: Stats
  - http://www.tinyos.net/stats.html
- NCSA/UIUC Internet Visualization Graphic
  - http://tinyurl.com/d7qgr

Related Work

- Carrier Sensing
  - B-MAC [Polastre04SenSys]: Make the packet preamble as large as the duty cycle
  - WiseMAC [ElHoiydi04Algosensors]: Send the packet preamble during the receiver's next scheduled CS time
  - We apply CS to synchronous protocols
- Dynamic Listening Periods
  - T-MAC [VanDam03SenSys]: Extends S-MAC to increase the listen time as data packets are received
  - DPSM/IPSM [Jung02Infocom]: Extends 802.11 for dynamic ATIM windows in single-hop environments
  - We use physical layer CS to work in multihop environments without inducing extra packet overhead

Properties of Preamble Sampling

- No synchronization necessary
  - We require synchronization
- Larger preambles increase chance of collisions
  - We restrict CS signals to a time when data is not being transmitted
  - In our technique, interference is tolerable between CS signals
- Broadcasts require preamble size be as long as a BI → Exacerbates broadcast storm
  - We do not require extra overhead for broadcast
- Only one sender can transmit to a receiver per BI
  - We allow multiple senders for a receiver per BI

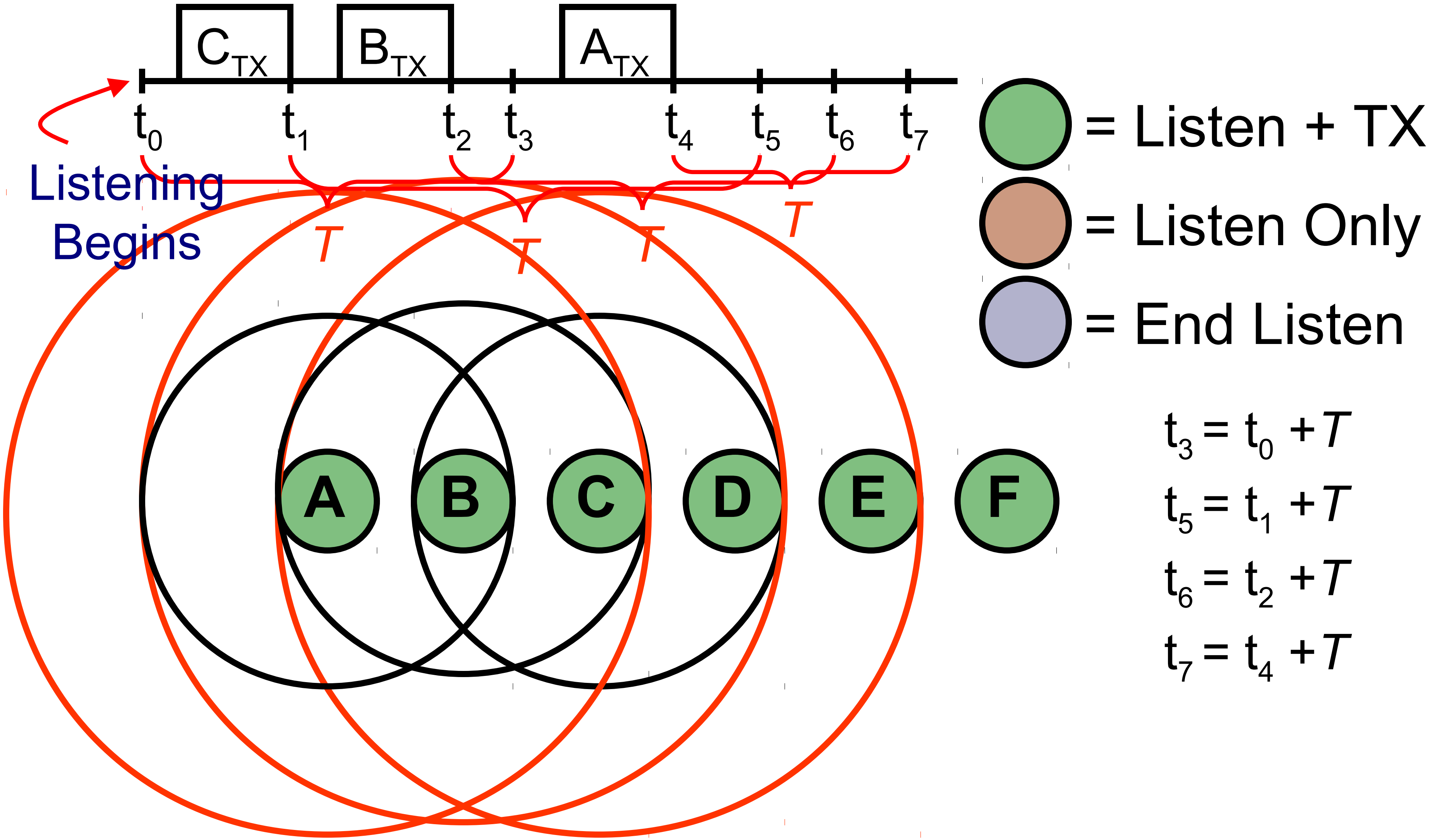
Is time synchronization a problem?

- Motes have been observed to drift 1 ms every 13 minutes [Stankovic01Darpa]
- The Flooding Time Synchronization Protocol [Maróti04SenSys] has achieved synchronization on the order of one microsecond
- Synchronization overhead can be piggybacked on other broadcasts (e.g., routing updates)
- GPS may be feasible for outdoor environments
- Chip scale atomic clocks being developed that will use 10-30 mW of power [NIST04]

Transition Costs Depend on Hardware [Polastre05IPSN/SPOTS]

Mote Radio Model	Wake-Up Time (ms)	TX/RX/ Sleep (mW)	Bitrate (kbps)
TR1000 (*1998-2001*)	0.020	36/12/ 0.003	40 ASK
CC1000 (*2002-2004*)	2	42/29/ 0.003	38.4 FSK
CC2420 (*2004-now*)	0.580	35/38/ 0.003	250 O-QPSK

Using Carrier Sensing for Adaptive Listening



$t_3 = t_0 + T$

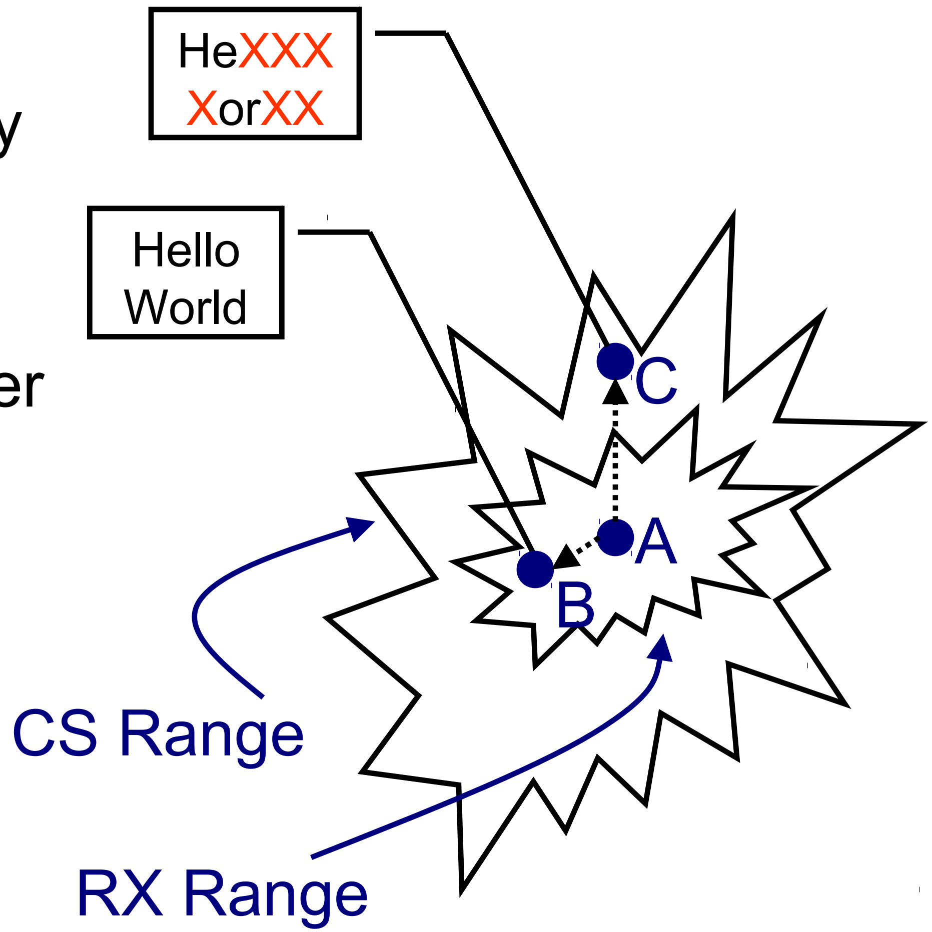
$t_5 = t_1 + T$

$t_6 = t_2 + T$

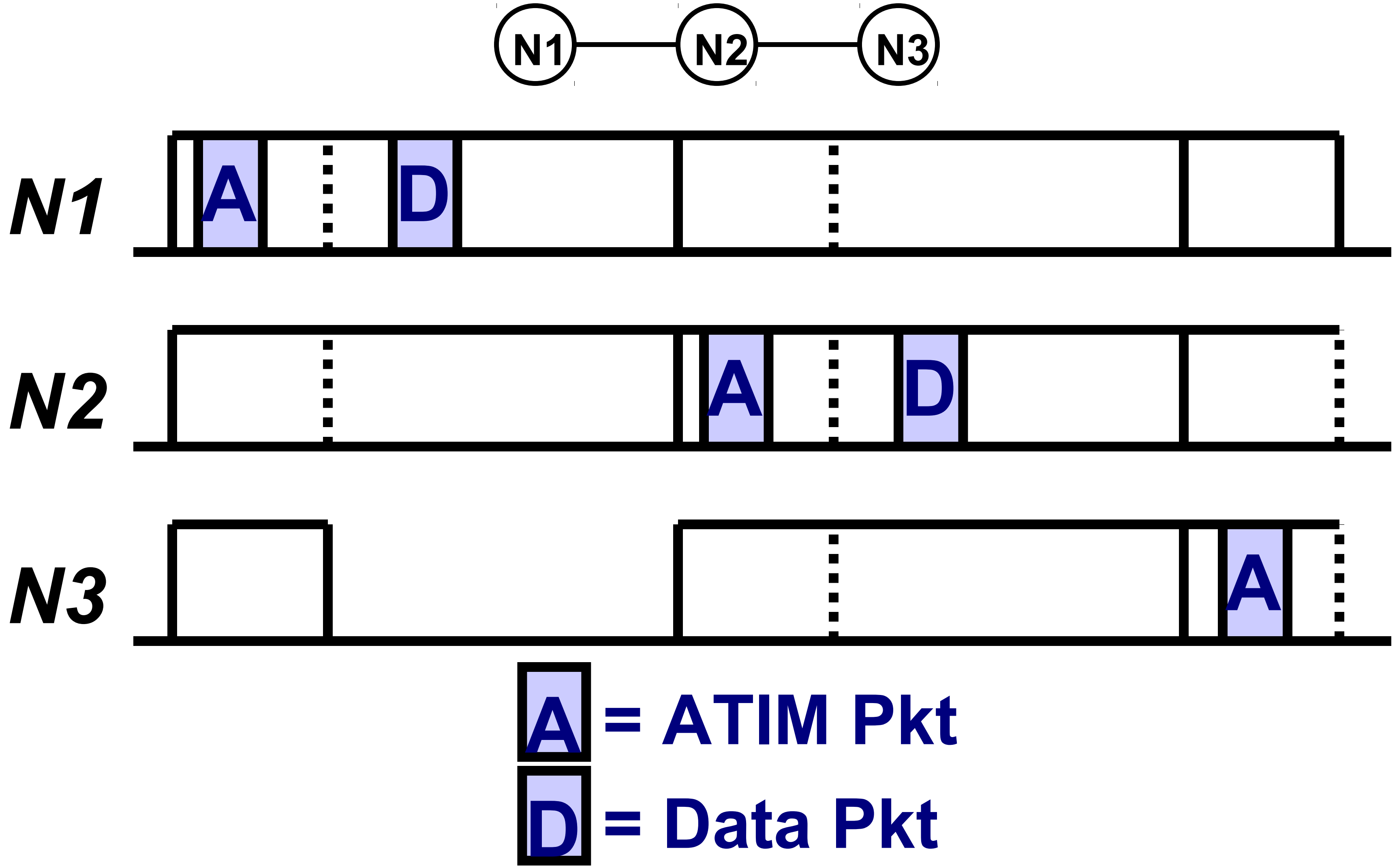
$t_7 = t_4 + T$

Adaptive Listening Background: RX Threshold vs. CS Threshold

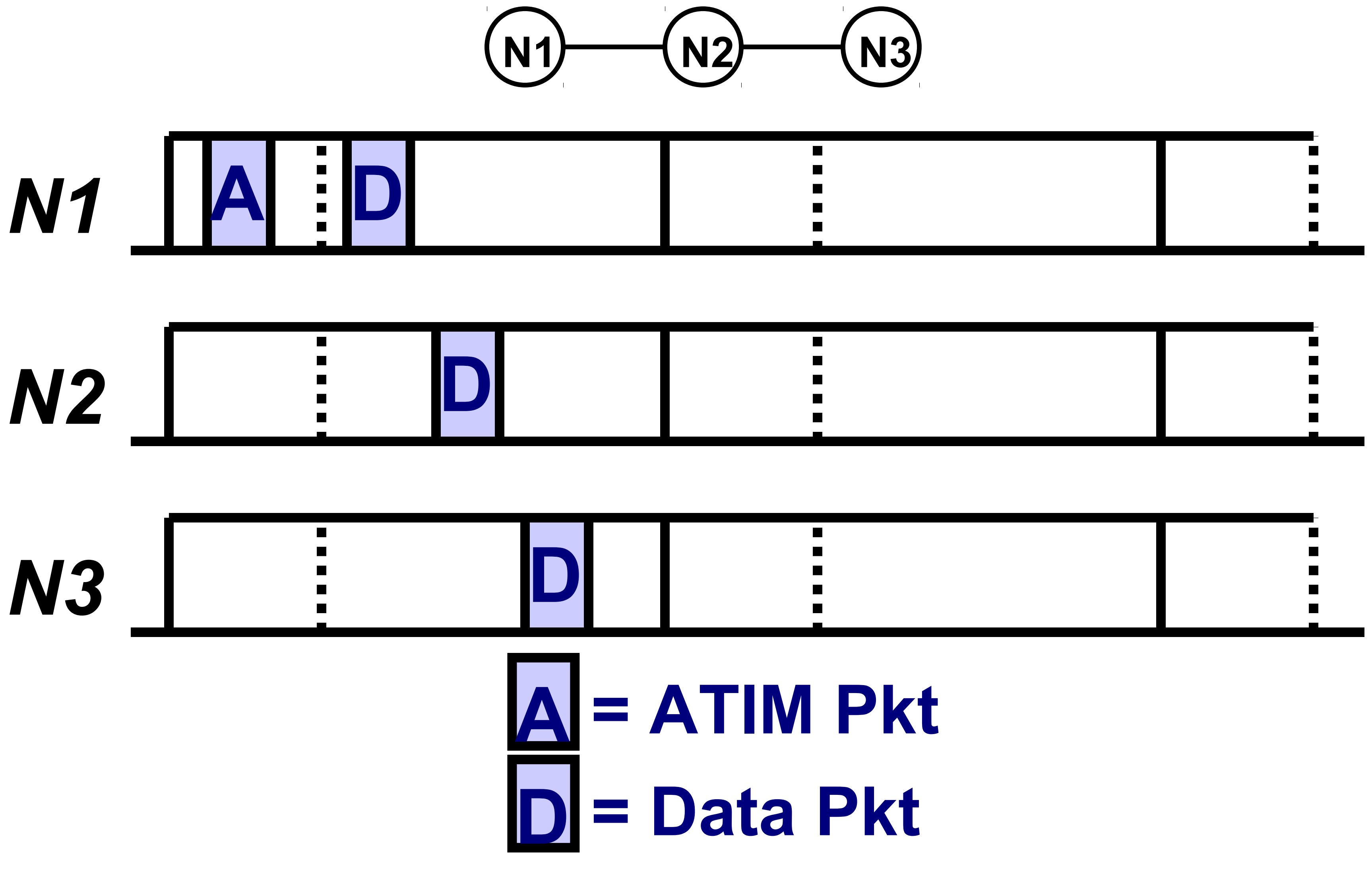
- RX Threshold: received signal strength necessary for a packet to be correctly received
- CS Threshold: received signal strength to consider the channel busy
- We assume that usually CS range ≥ 2*RX range
  - If this is not true, our technique gracefully degrades to a fixed listening interval scheme



Protocol Extreme #1



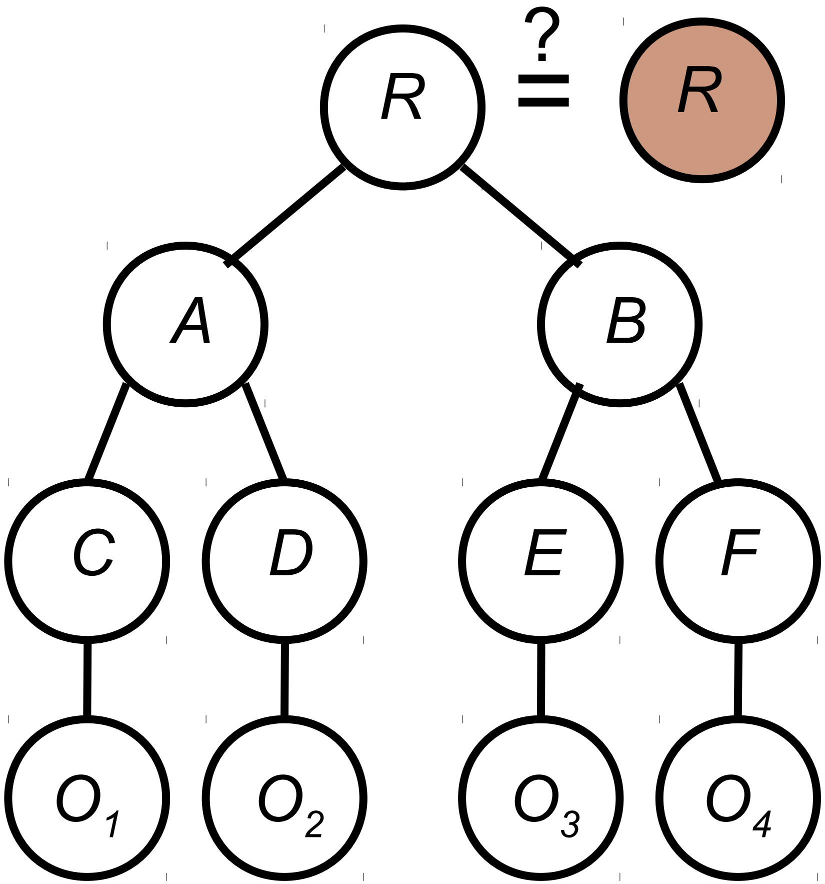
Protocol Extreme #2



Wireless Channel Diversity

- Radios typically have multiple non-interfering, half-duplex channels
  - 802.11b: 3 channels
  - 802.11a: 12 channels
  - Zigbee (used on Telos motes): 16 channels
- At any given time, an interface can listen to at most one channel

Merkle Tree Authentication



$C = \text{hash}(O_1)$

$A = \text{hash}(C \,||\, D)$

$R = \text{hash}(A \,||\, B)$

Each sensor given *R* and O(lg *N*) other hashes