Using Channel Diversity to Improve Security for Wireless Sensor Networks

Matthew J. Miller
University of Illinois
at Urbana-Champaign

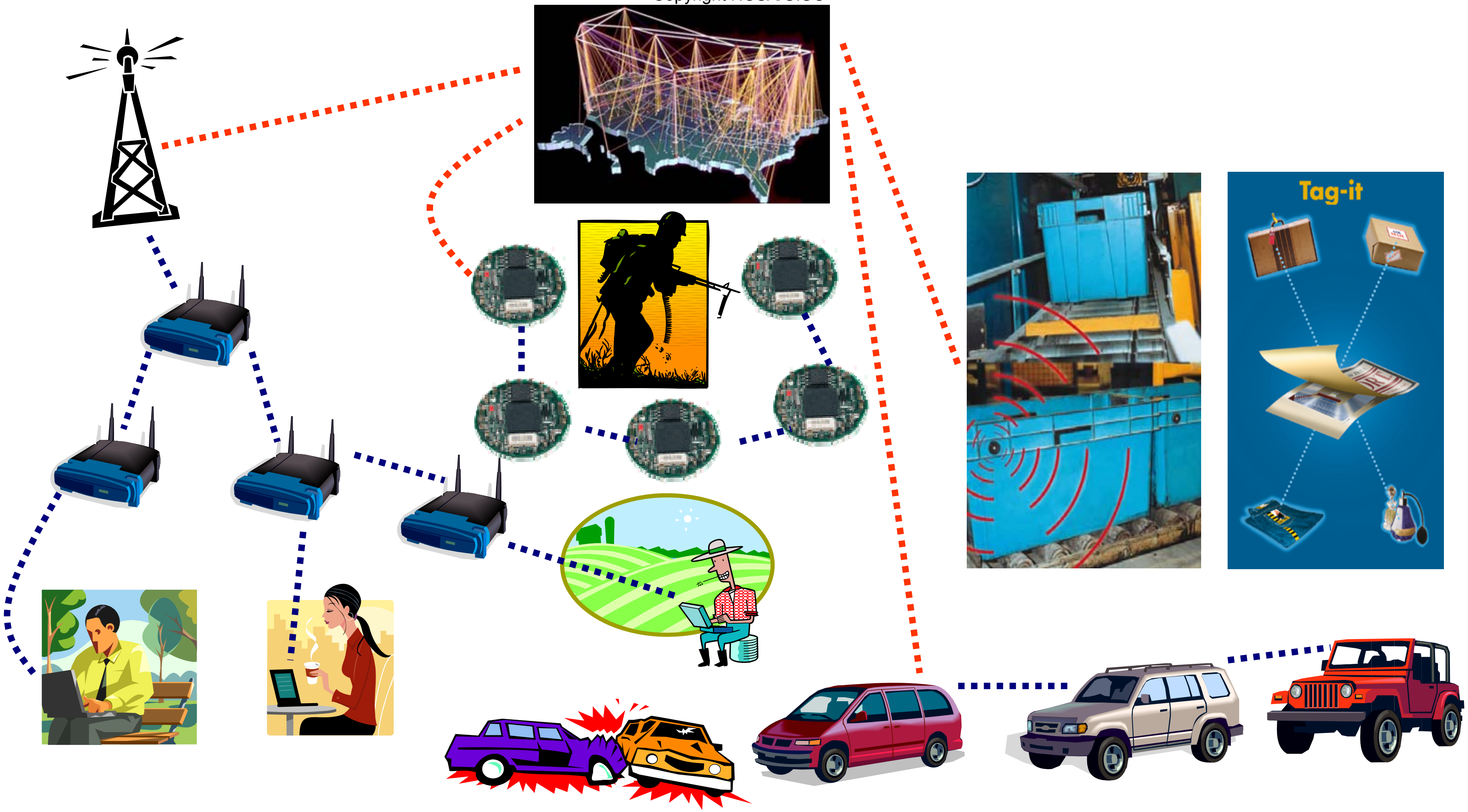
Wireless Networking: It's Kind of a Big Deal

- "***The number of WiFi hotspots in the United States*** increased from 3,400 to 21,500 between 2002 and 2004 […] that number is expected to grow […] to 64,200 by 2008, ***a 31.5 percent compound annual growth rate***." – David A. Gross,

  US Ambassador Bureau of Economic and Business Affairs
- "***The number of RFID tags produced worldwide is expected to increase more than 25 fold*** between 2005 and 2010, reaching 33 billion, according to market research company In-Stat." – EE Times
- "IDC now estimates there will be more than ***100 million Bluetooth devices worldwide*** by the end of the year, and In-Stat/MDR expects a compound ***annual growth rate of 60 percent*** from 2003 to 2008." – CNET.com
- ***TinyOS*** Sensor Operating System: Typically ***50-200 downloads per day*** – TinyOS Website

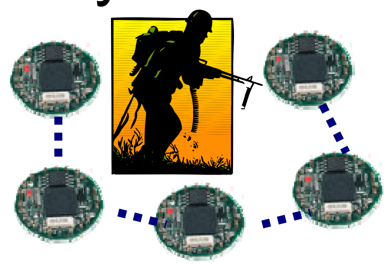
Emerging Wireless Applications

Copyright NCSA/UIUC



Why Use Multihop Wireless?

- Connectivity: Extend infrastructure at a low cost
  - Mesh and community Networks
- Ease of Deployment: Extend infrastructure quickly
  - Disaster scenarios
  - Sensor networks
  - Vehicular networks
  - Military operations



- Performance: Increased capacity per node

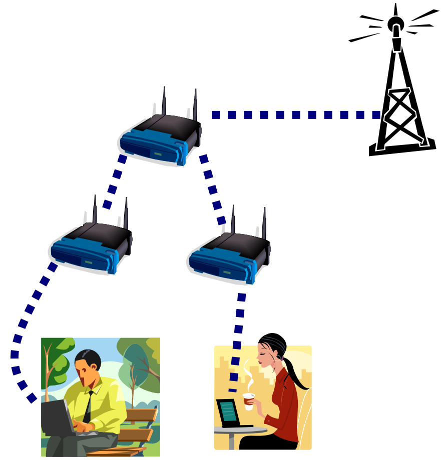
  (W = Channel bitrate, N = Number of nodes)

Single Hop Network

$$O\left(\frac{W}{N}\right)$$

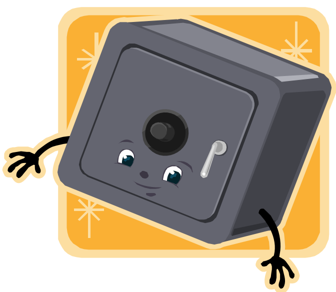
Multihop Network [Gupta00Capacity]

$$O\left(\frac{W}{\sqrt{N}}\right)$$



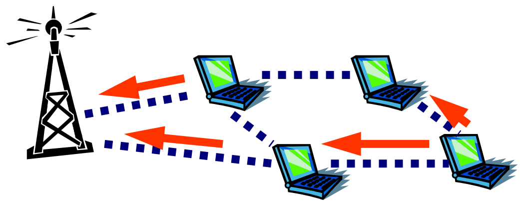
Some Research Challenges

- Improve performance
  - Exploit diversity (e.g., multiple channels, bitrates)
- Security and privacy
  - Resource constraints on cryptography
  - Tapping the channel to eavesdrop is much easier
- Energy efficiency
  - The power cable has proved remarkably resilient in this “wireless” world



Summary of My Work

- Exploiting channel diversity for secure key distribution in sensor networks
- Adaptive energy efficient protocols for wireless devices
- Protocol implementations
  - Power save broadcast on sensors (TinyOS)
  - User-level ad hoc routing protocol in Linux



Talk Outline

- Background on Wireless Sensor Network Key Distribution
- Leveraging Channel Diversity for Key Distribution
- Adaptive Energy-Saving Protocols
- Future Research

Talk Outline

- Background on Wireless Sensor Network Key Distribution
- Leveraging Channel Diversity for Key Distribution
- Adaptive Energy-Saving Protocols
- Conclusion

Key Distribution Problem Statement

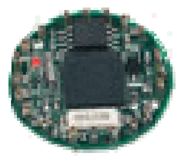
- After deployment, a sensor needs to establish pairwise symmetric keys with neighbors it discovers for confidential and authenticated communication

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

Design Considerations for Wireless Sensor Networks

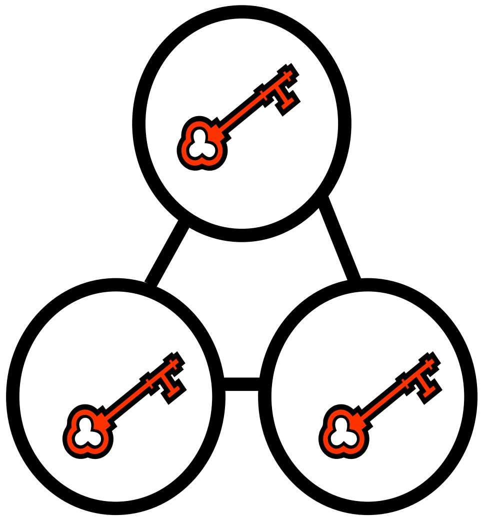
- Resource constrained
  - Energy, computation, memory, bitrate

- Large scale deployments
  - May need thousands (or more) of devices
- Topology may be uncontrolled
  - Specific device's location unknown in advance

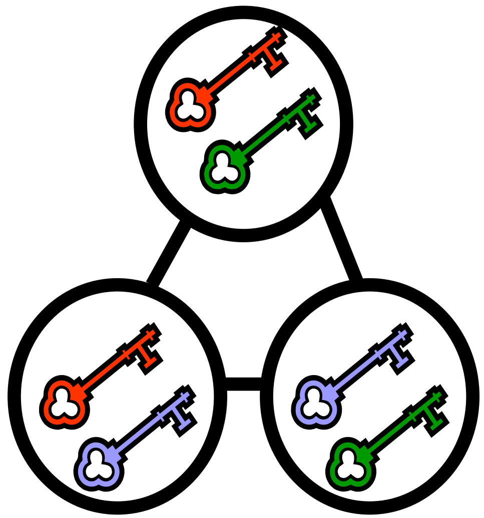


Design Space

- Every sensor 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 sensor pair
  - ☺ One compromised node does not affect the security of any other links
  - ☹ Required sensor storage scales linearly with network size



Outline of Solution Approaches

- 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

Outline of Solution Approaches

- 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?

Outline of Solution Approaches

- 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

Talk Outline

- Background on Wireless Sensor Network Key Distribution
- Leveraging Channel Diversity for Key Distribution
- Adaptive Energy-Saving Protocols
- Future Research

Talk Outline

- Background on Wireless Sensor Network Key Distribution
- Leveraging Channel Diversity for Key Distribution
- Adaptive Energy-Saving Protocols
- Conclusion

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

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 $(1/c)^M \rightarrow 0$:

  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

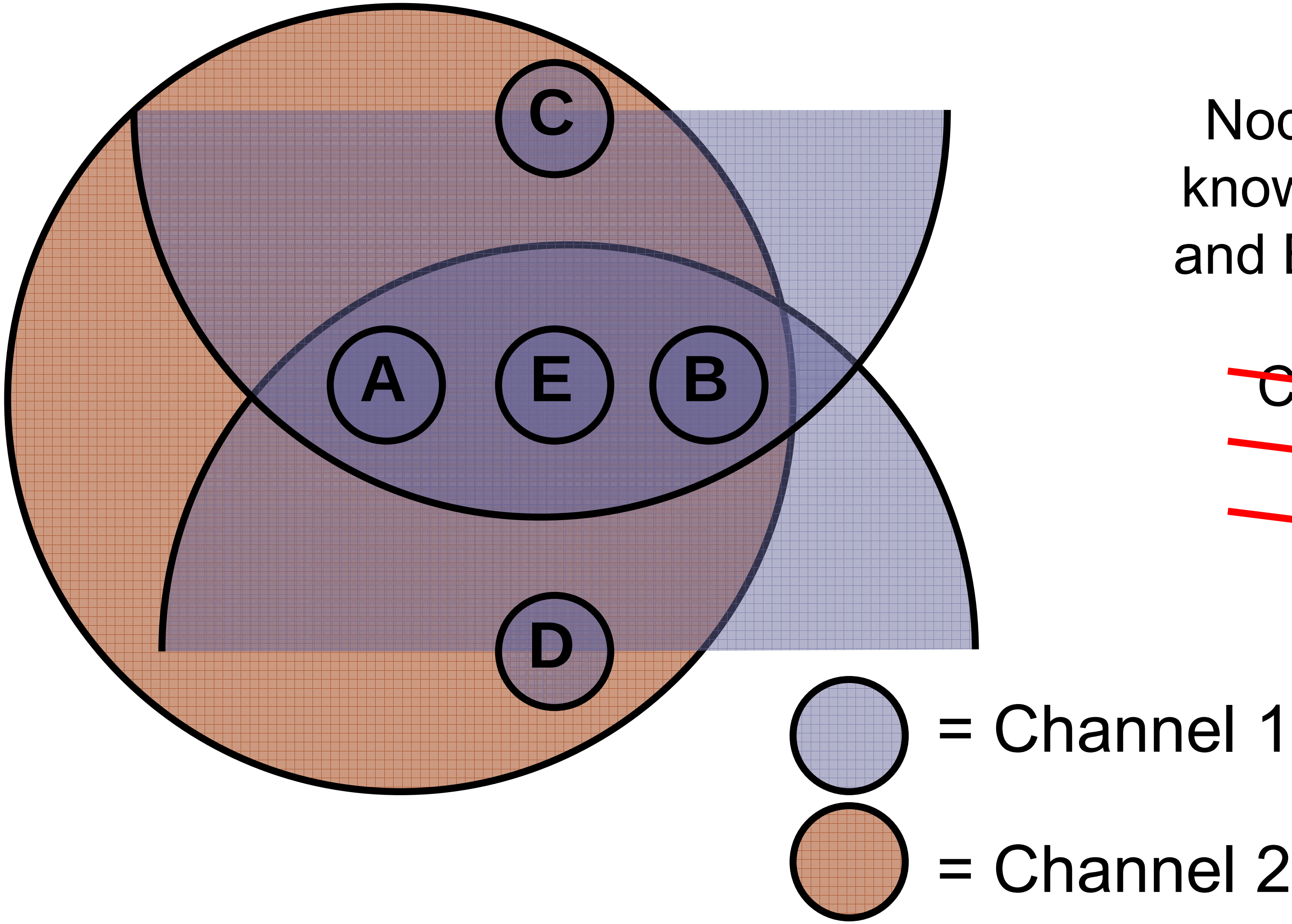
Phase 1: Predeployment

- 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



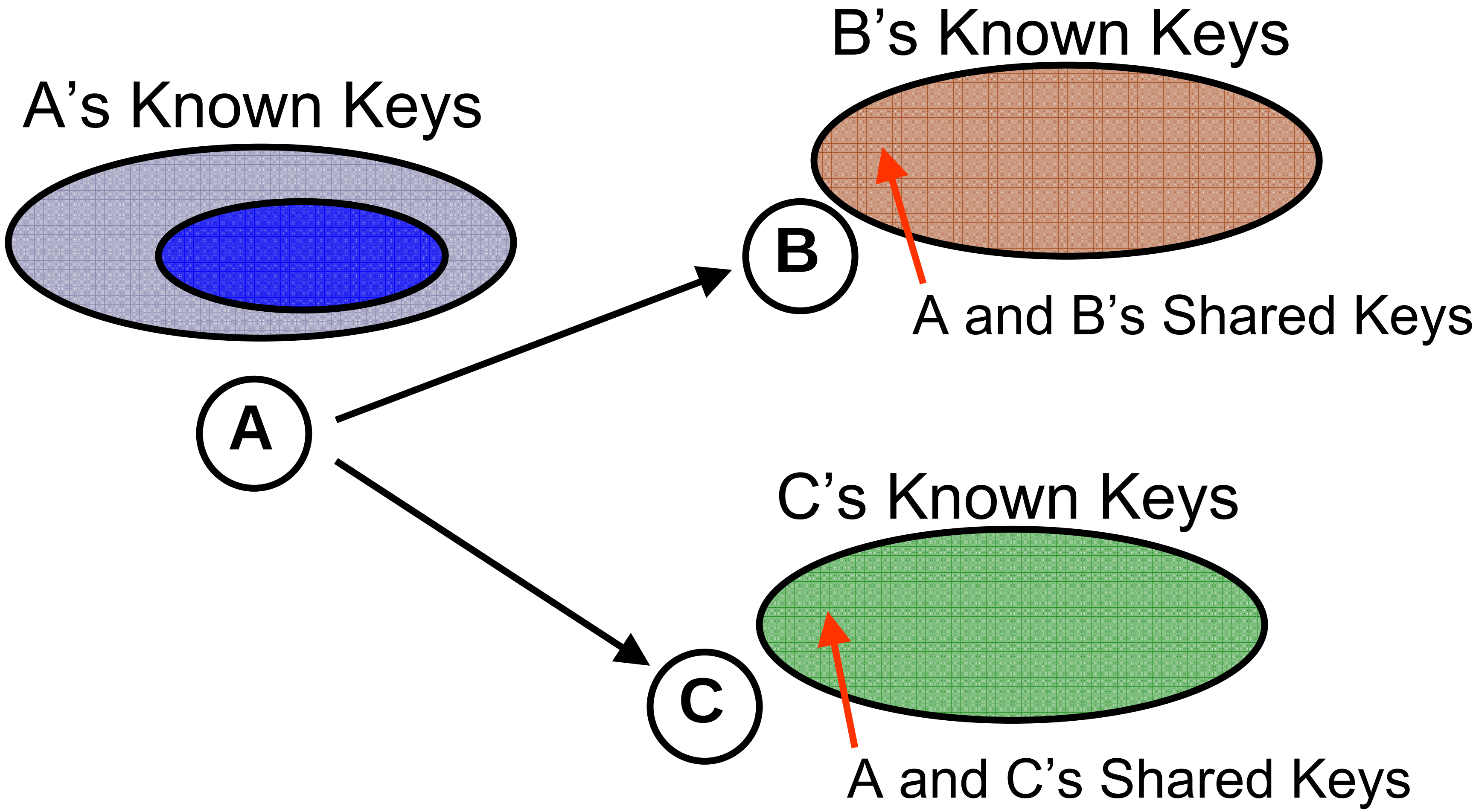
Nodes that know all of A and B's keys:

~~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

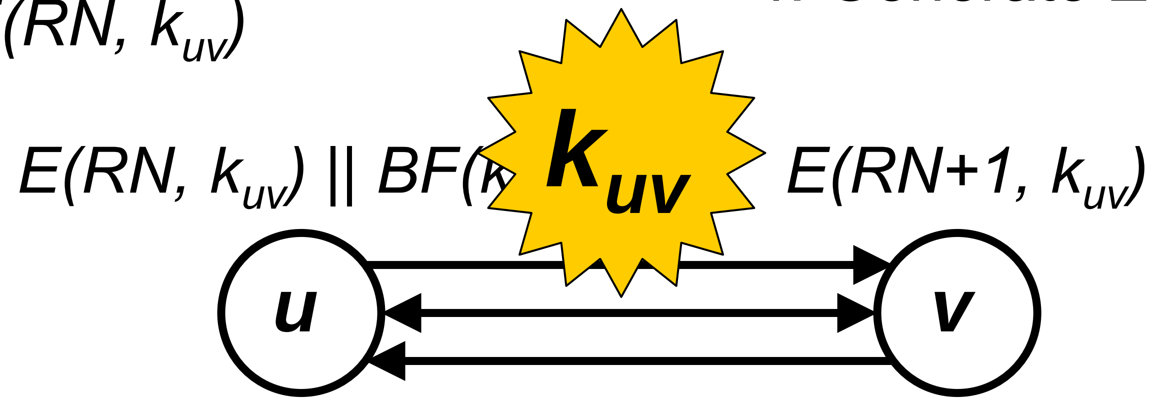


Phase 4: Key Establishment

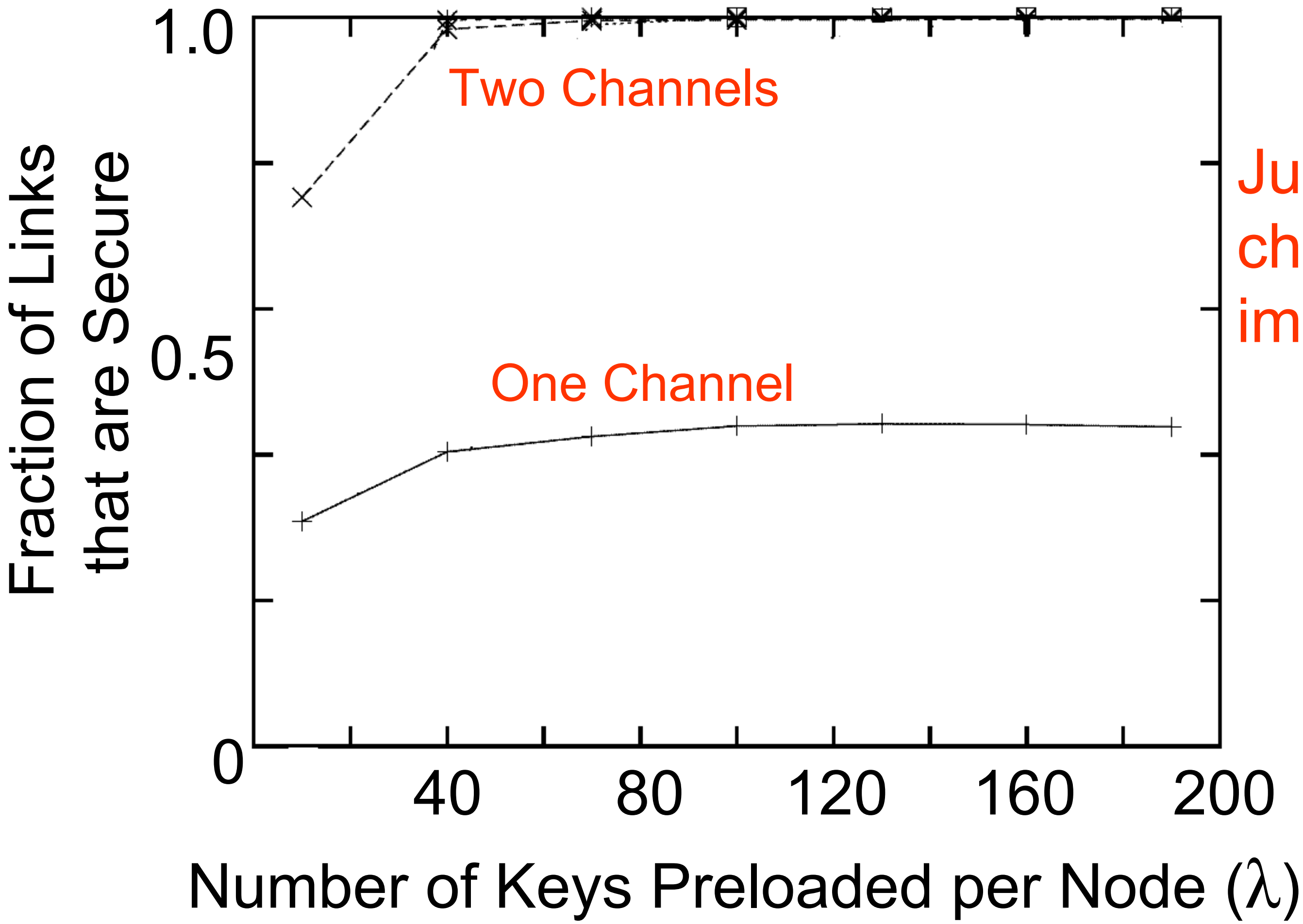
u's believed set of shared keys with $v = \{k_1, k_2, k_3\}$

1. Generate link key: $k_{uv} = \text{hash}(k_1 \| k_2 \| k_3)$
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 $E(RN+1, k_{uv})$

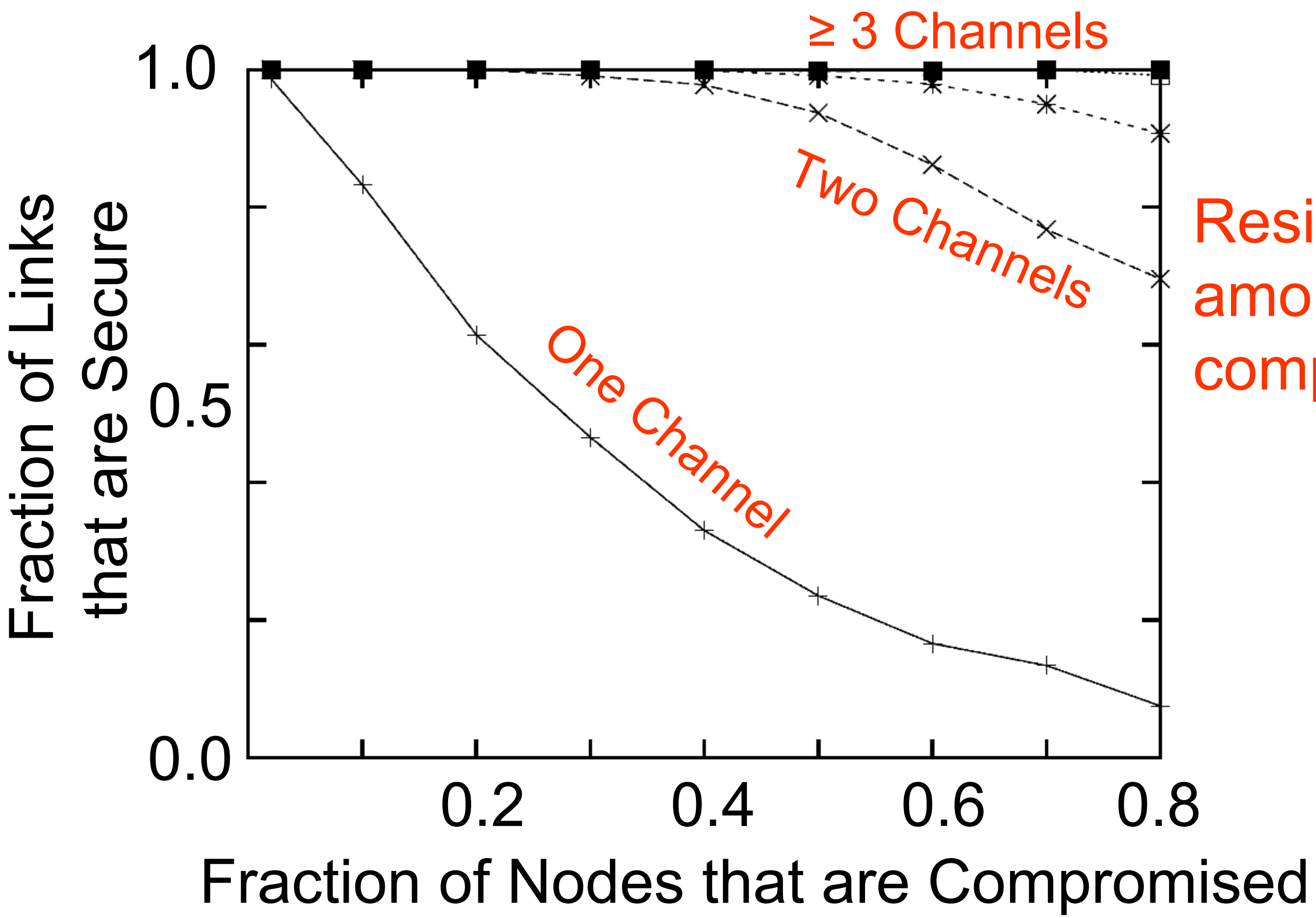


Simulation Results



Just one extra channel significantly improves security

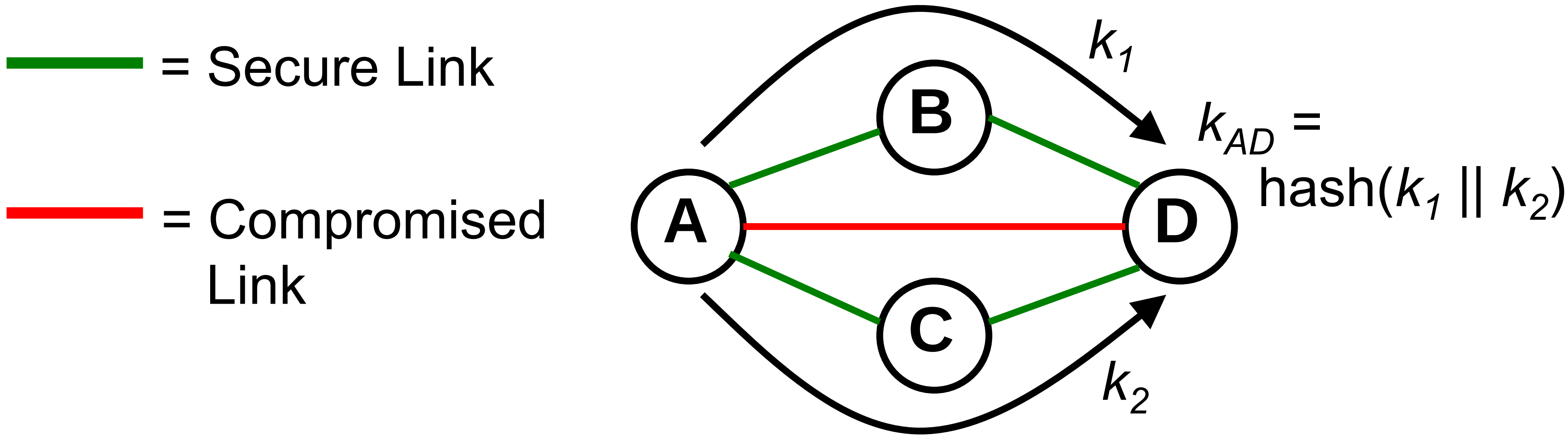
Simulation Results



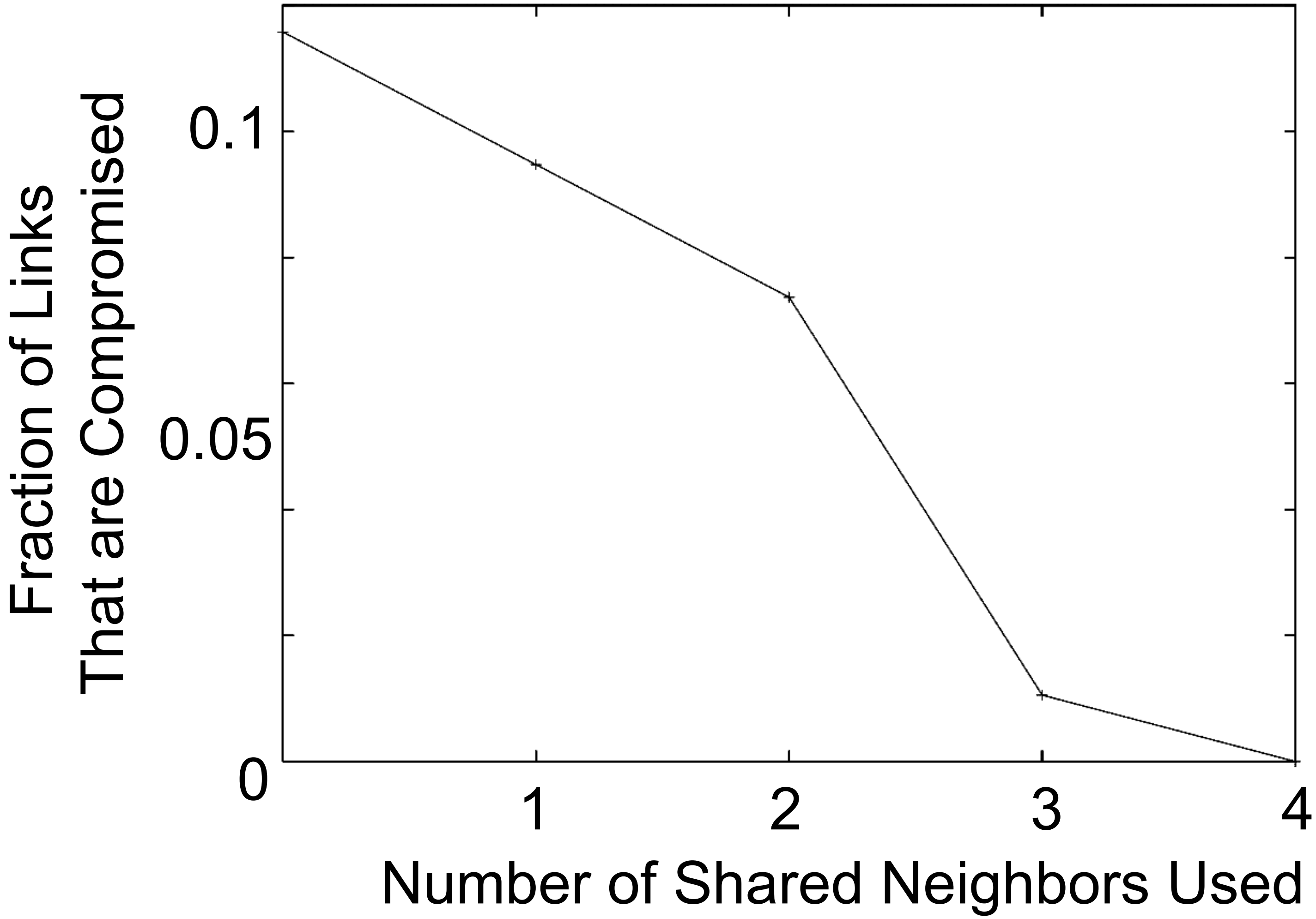
Resilient to large amount of node compromise

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



Key Distribution 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

Talk Outline

- Background on Wireless Sensor Network Key Distribution
- Leveraging Channel Diversity for Key Distribution
- Adaptive Energy-Saving Protocols
- Future Research

Talk Outline

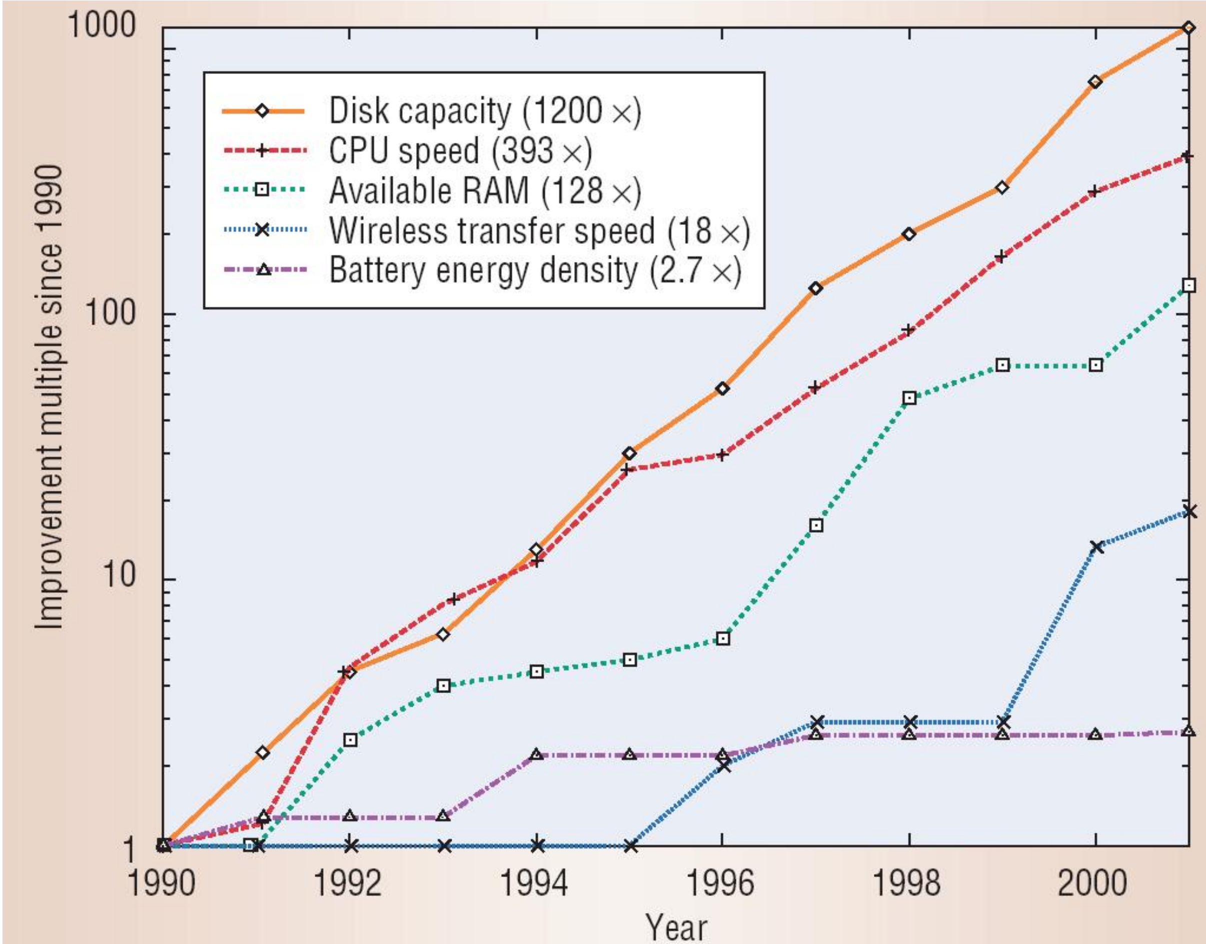
- Background on Wireless Sensor Network Key Distribution
- Leveraging Channel Diversity for Key Distribution
- Adaptive Energy-Saving Protocols
- Conclusion

Won't Moore's Law Save Us?

NO!!!

Necessitates Energy-Saving Protocol Design

Log Scale



1200 x

393 x

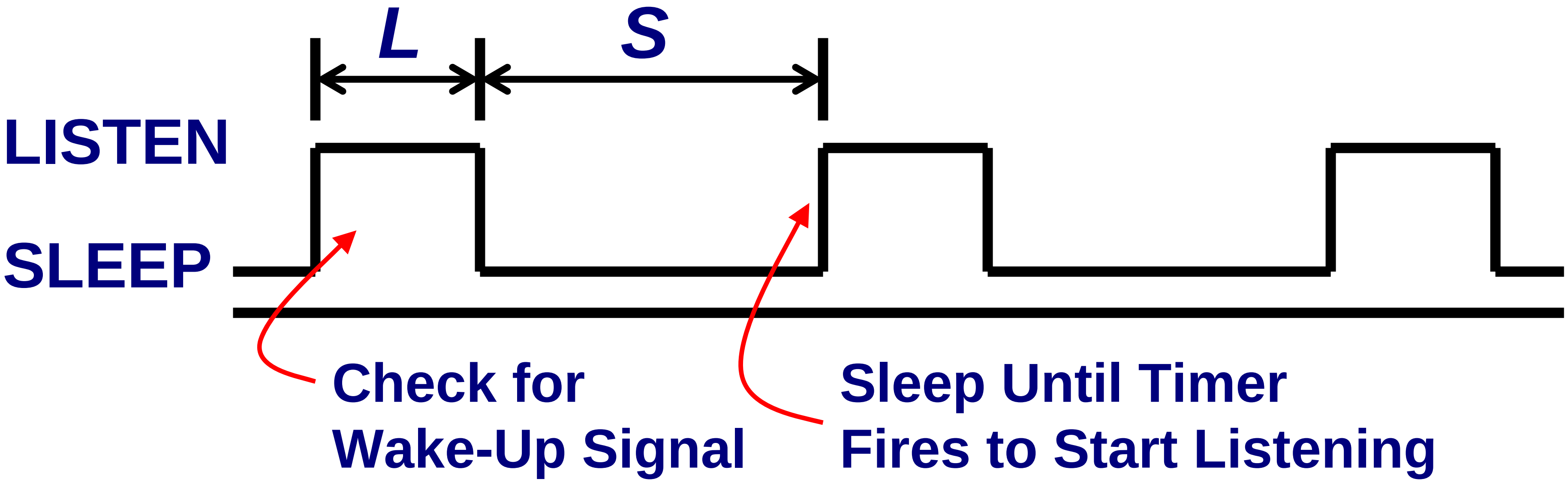
128 x

18 x

2.7 x

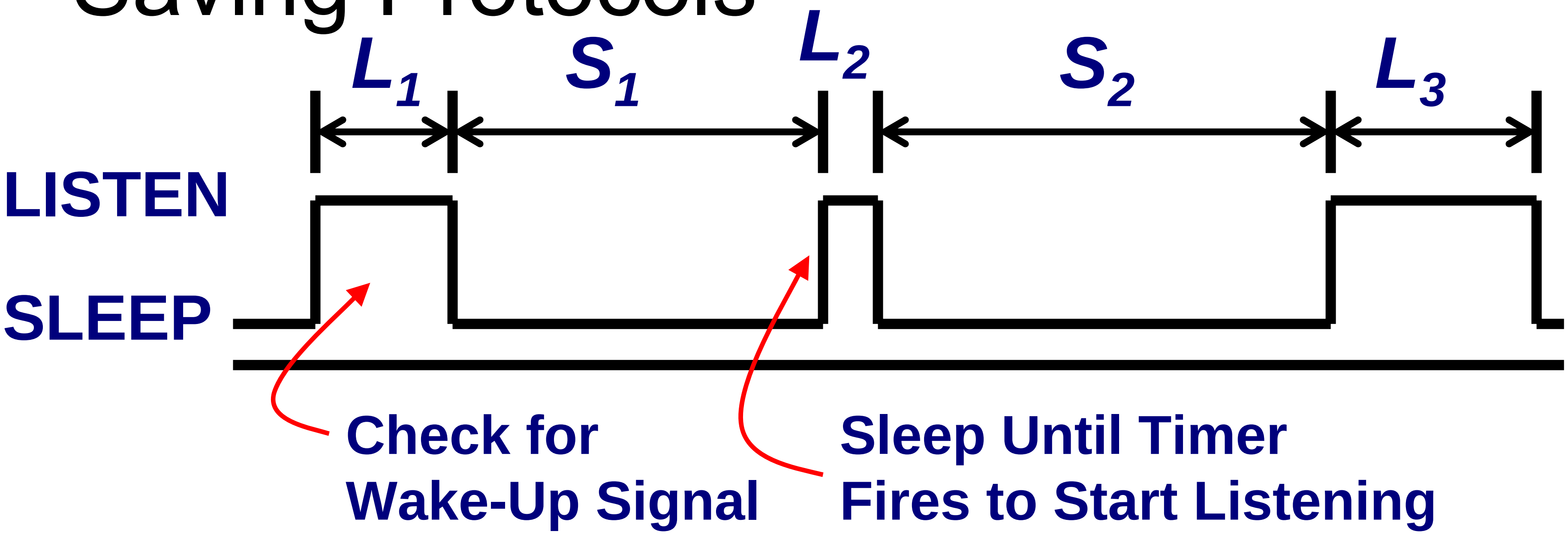
From "Thick Clients for Personal Wireless Devices" by Thad Starner in *IEEE Computer*, January 2002

Common Power Save Protocol Design



- *L* and *S* are ***static*** values regardless of traffic
- Design used in IEEE 802.11 as well as sensor protocols (e.g., B-MAC and STEM)

Our Approach: Adaptive Energy-Saving Protocols

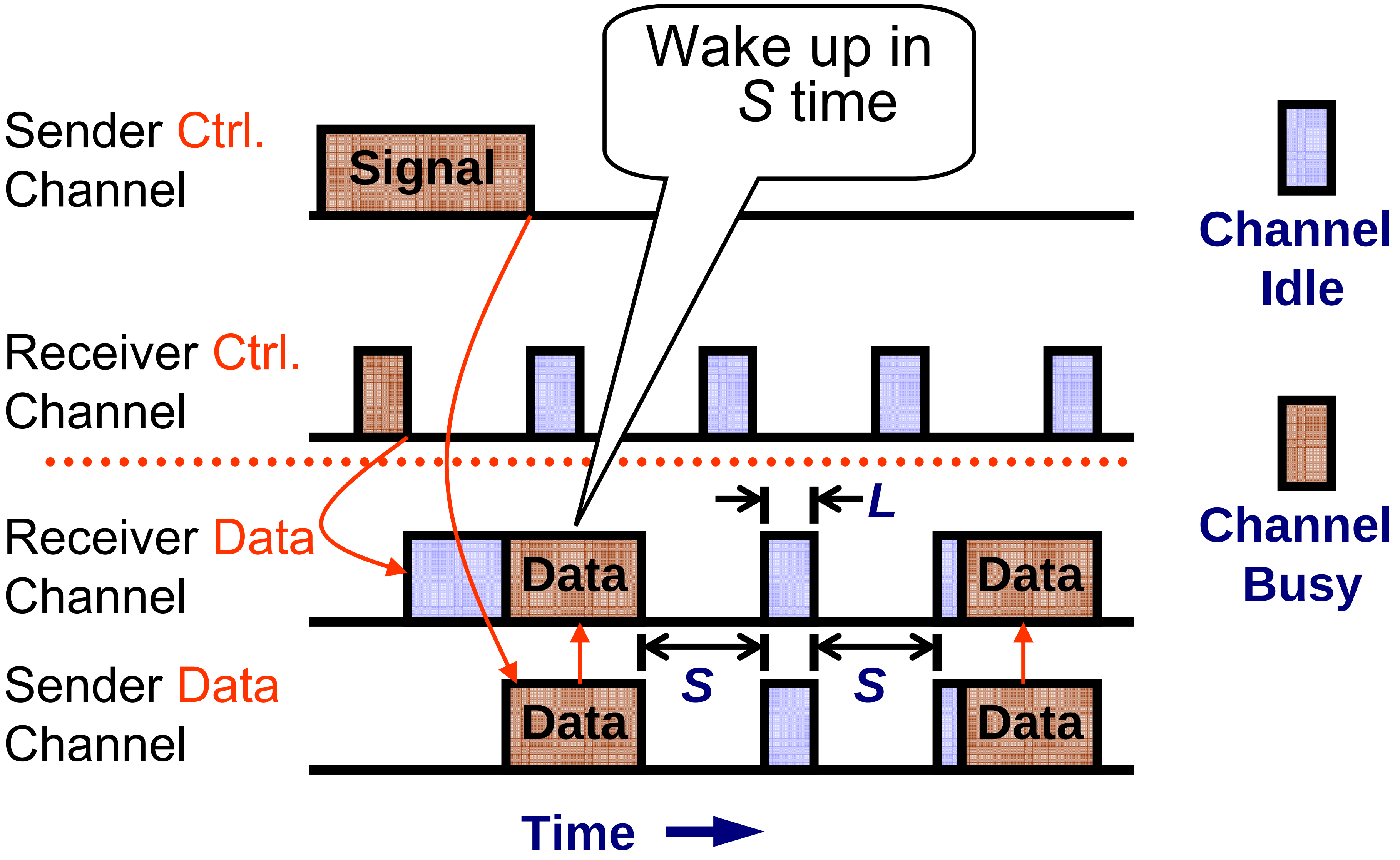


- Adapt listening (*L*) based on channel state
- Adapt sleeping (*S*) based on traffic arrivals and desired latency

Protocol Design Space

	Adaptive Listening	**Adaptive Sleeping**
In-Band	Our MASS 2005 paper	Our multilevel routing work
Out-of-Band	Our in-band techniques are applicable	Covered in this talk

Out-of-Band Protocol Example



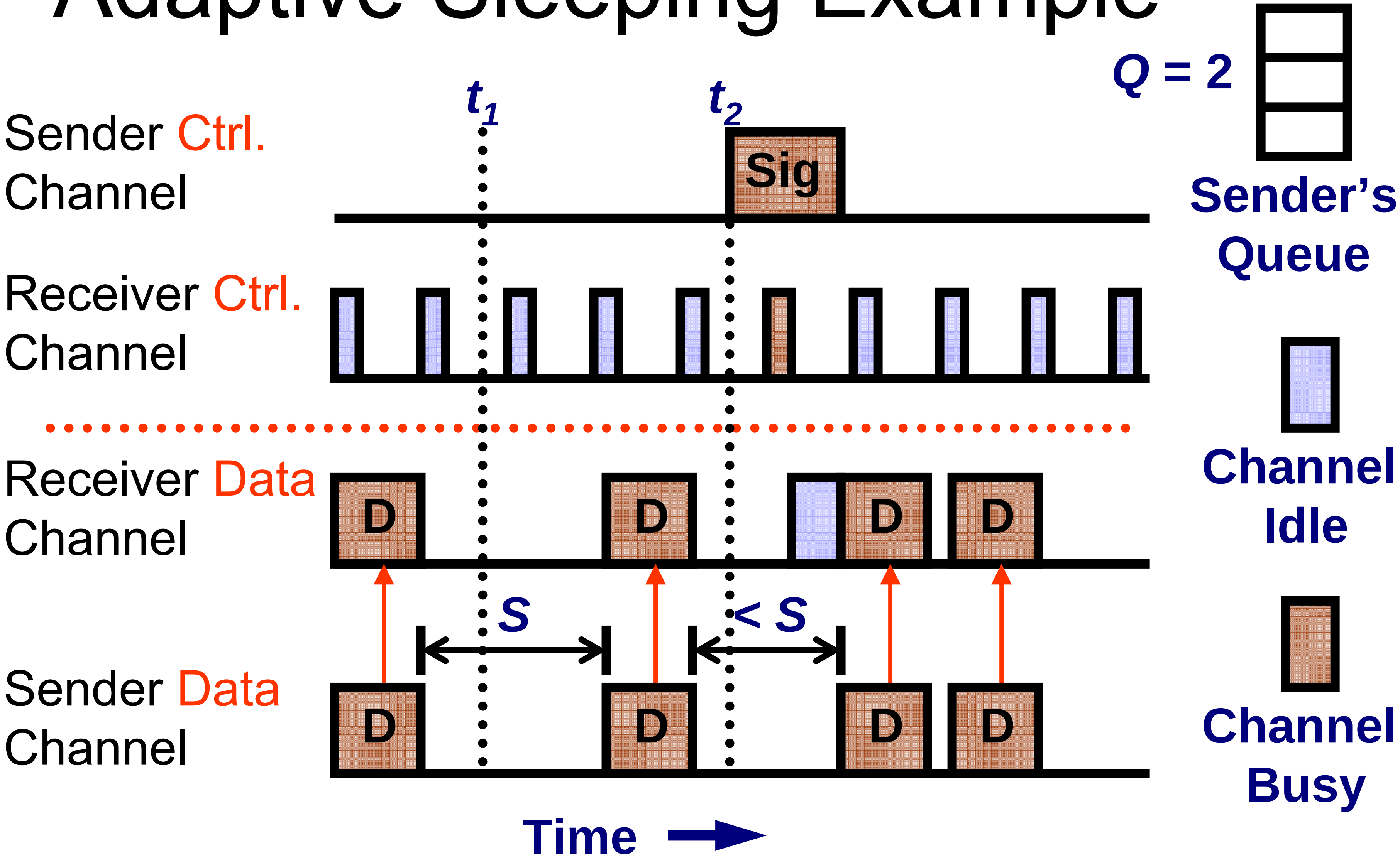
How Do You Choose *S*?

- If energy is our only concern then *S* can be arbitrarily large
  - However, the queue may become large
- Since sensors are resource limited, we address this queue constraint
  - If a device’s queue reaches a threshold, *Q*, then it must start transmitting packets soon

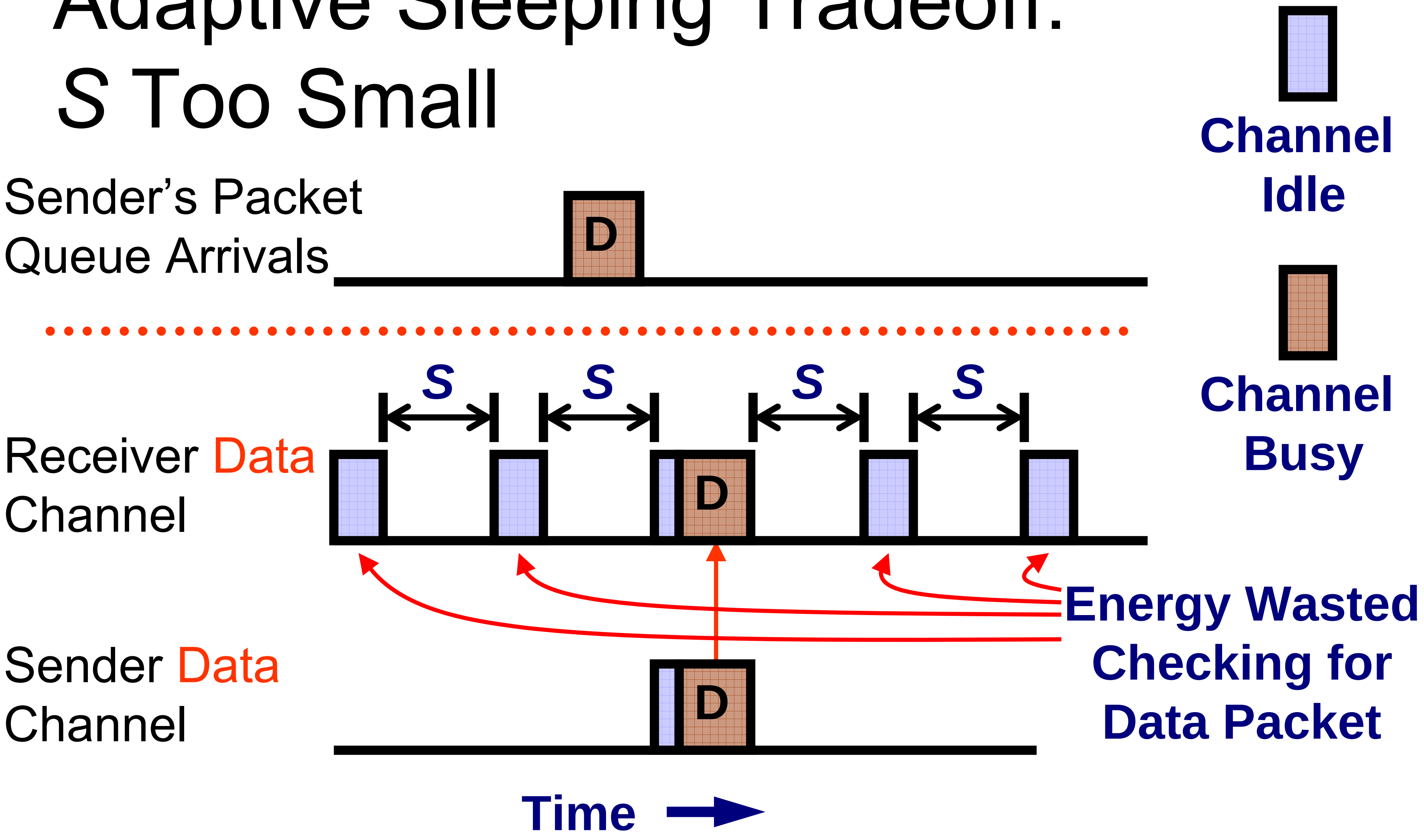
Adaptive Sleeping Overview

- Sender and receiver schedule a future wake-up time based on the traffic rate
- If the sender's queue reaches *Q* packets before a scheduled wake-up:
  - Then the sender wakes up the receiver via the out-of-band control channel
- All nodes periodically check control channel for wake-up signal
  - If signal detected → Turn on data radio
  - If data packet is for another node → Data radio returns to sleep

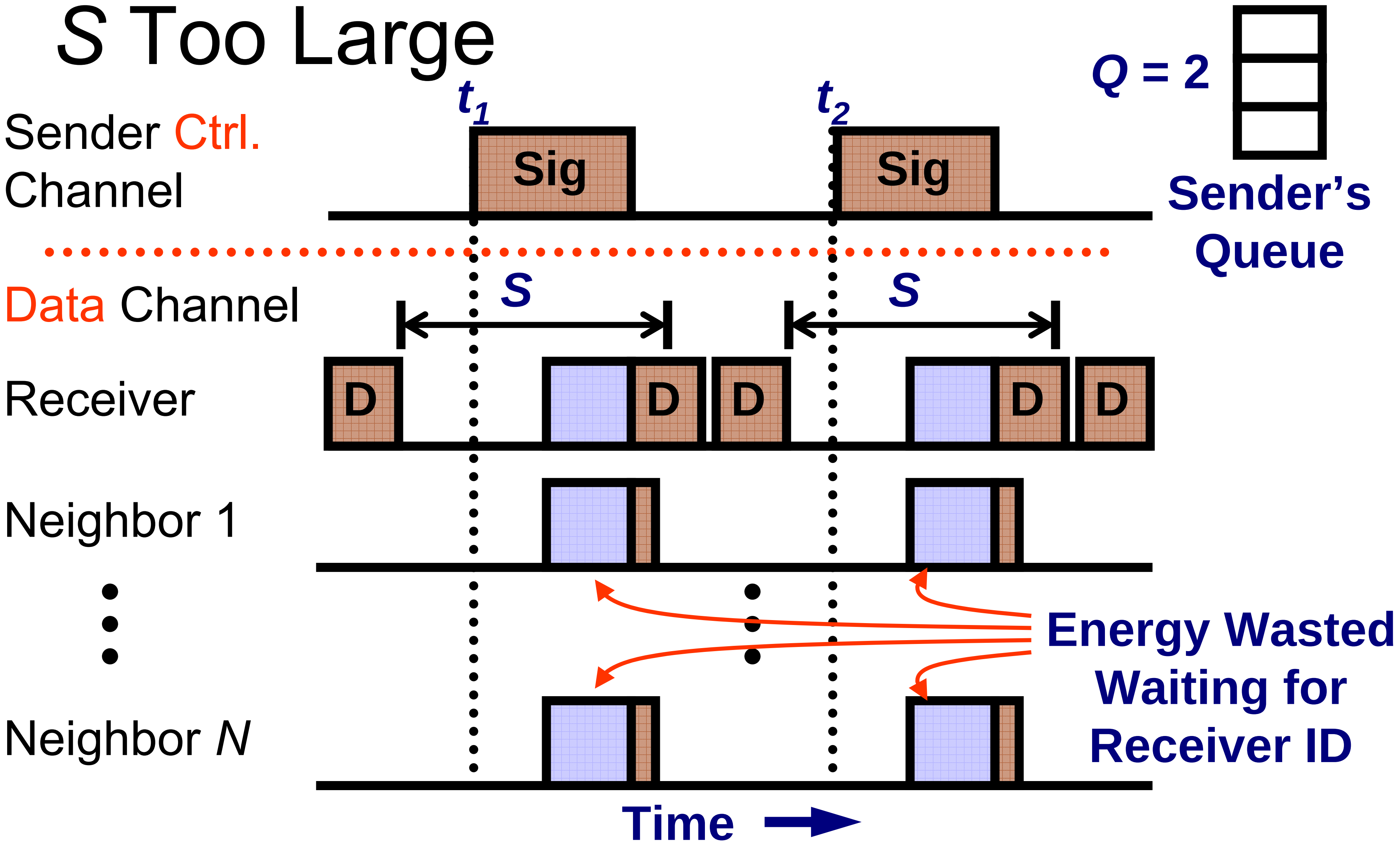
Adaptive Sleeping Example



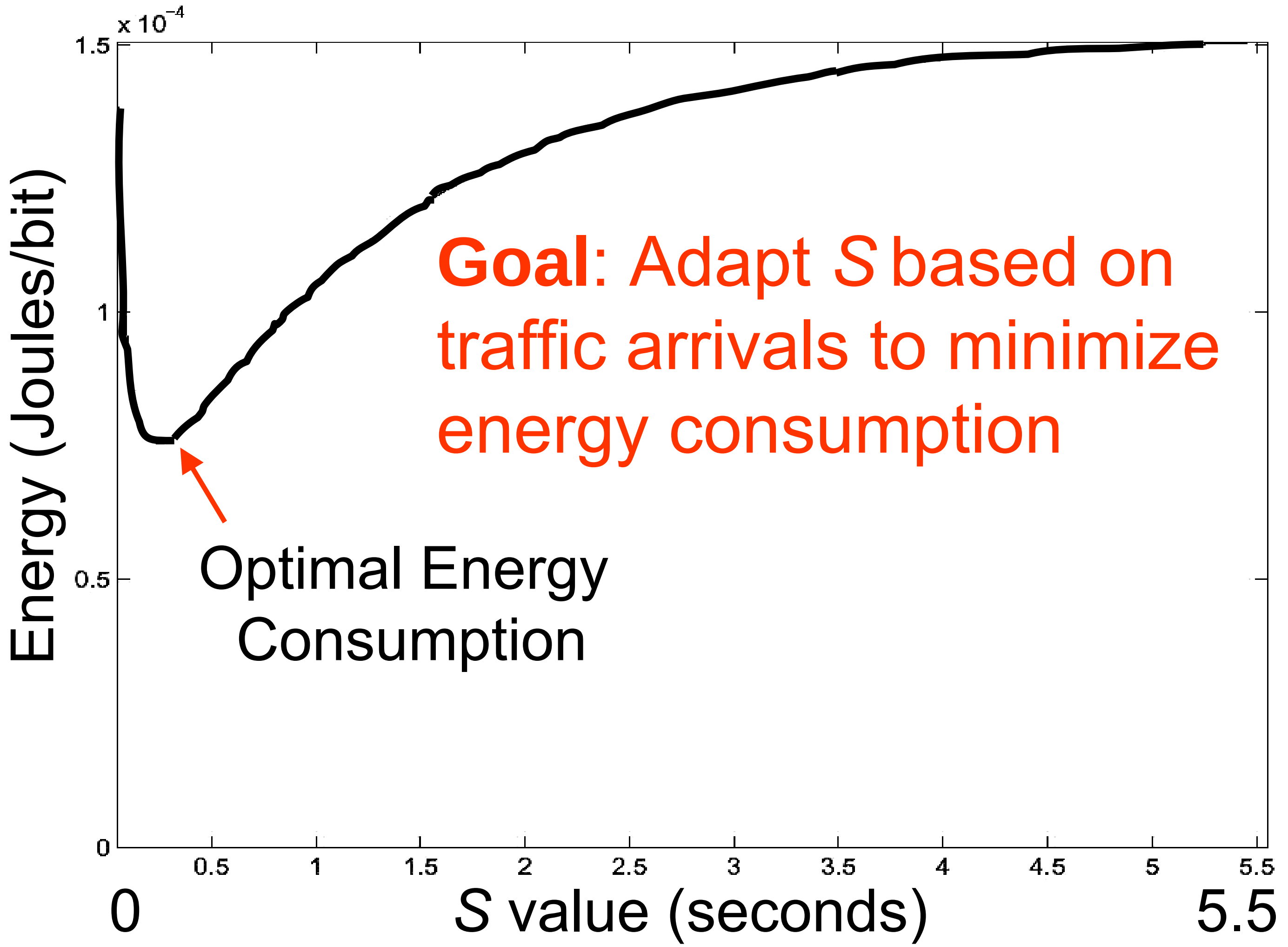
Adaptive Sleeping Tradeoff: *S* Too Small



Adaptive Sleeping Tradeoff: *S* Too Large



Adaptive Sleeping Tradeoff



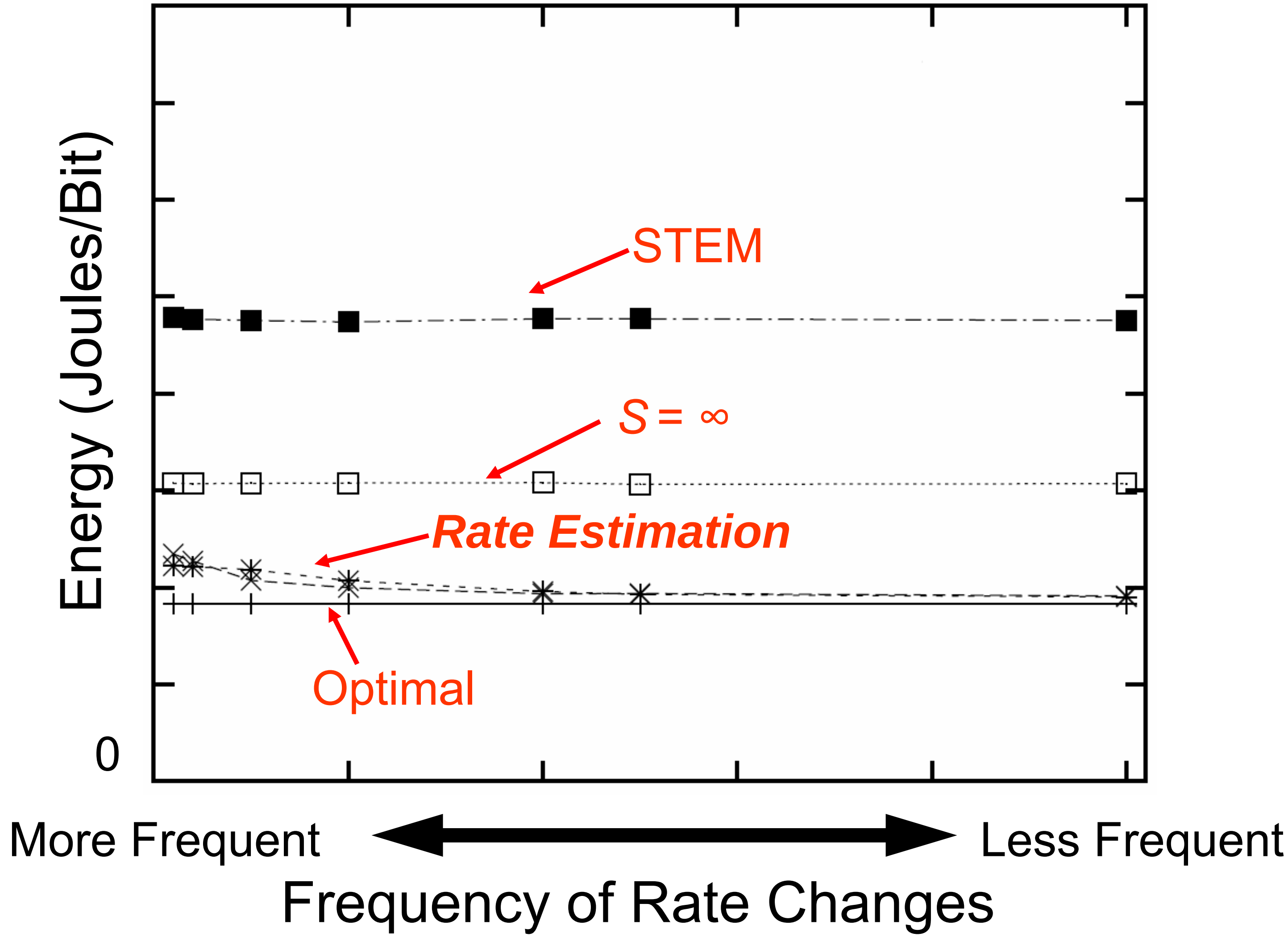
Adaptive Sleeping Analysis

- Based on analysis, we found that S is optimized according to the equation:

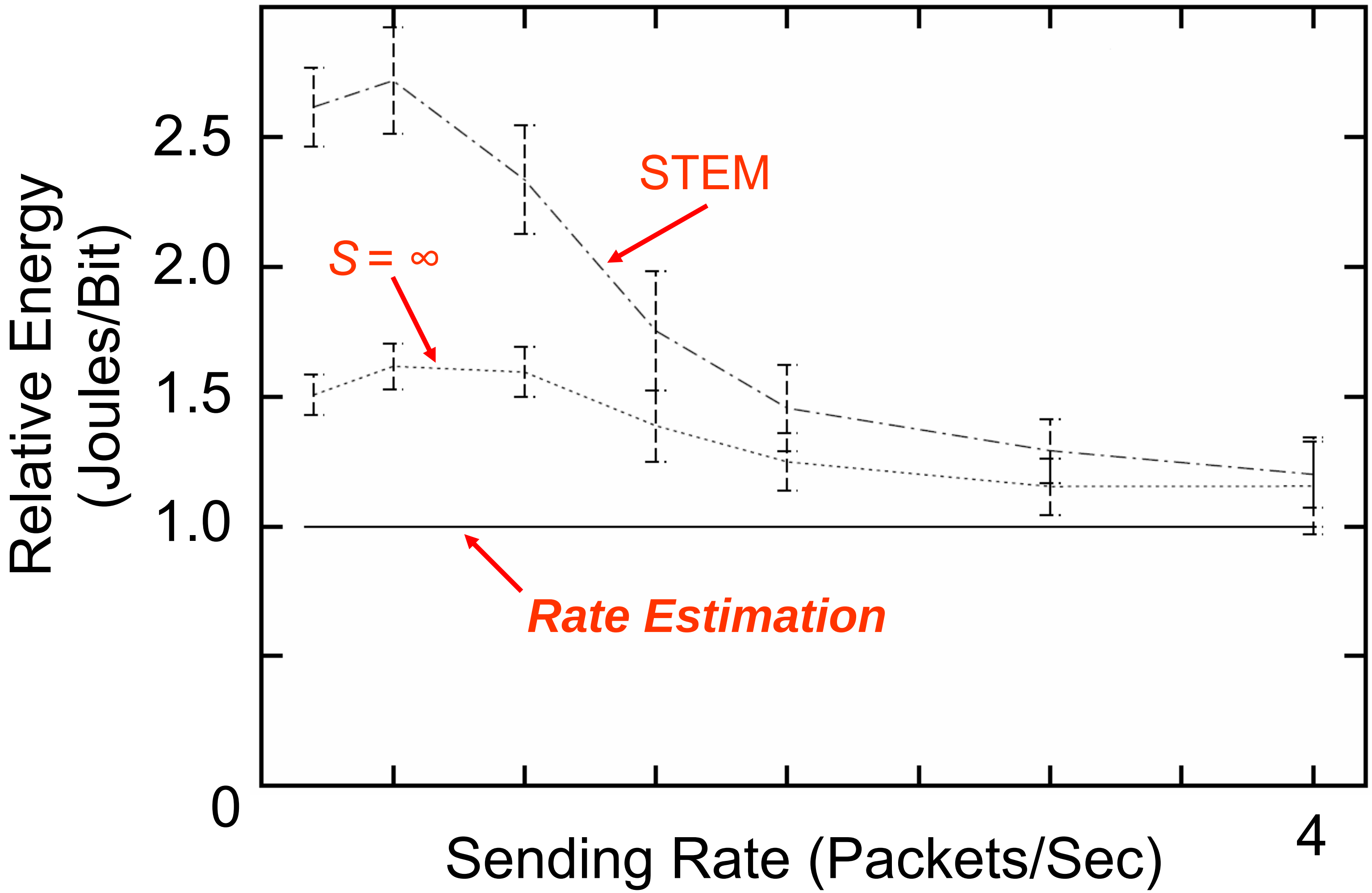
$$S = \gamma\,(1/R)$$

- R = Packet arrival rate at sender
  - Can be estimated with a weighted moving average
- γ = Function of Q and the number of neighbors of the sender (*nbrs*)
  - Can be calculated offline when Q and *nbrs* are known

Adaptive Sleeping: Time-Varying Traffic Rate Results



Adaptive Sleeping: Multihop Topology Results



Talk Outline

- Background on Wireless Sensor Network Key Distribution
- Leveraging Channel Diversity for Key Distribution
- Adaptive Energy-Saving Protocols
- Future Research

Talk Outline

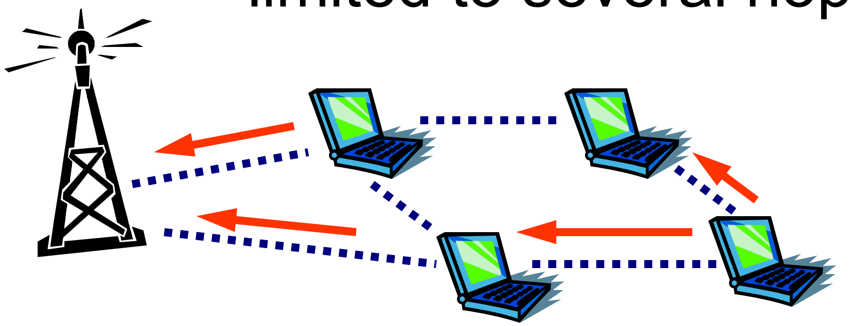
- Background on Wireless Sensor Network Key Distribution
- Leveraging Channel Diversity for Key Distribution
- Adaptive Energy-Saving Protocols
- Conclusion

Future Research: Multihop Wireless Networks

- Performance
  - Efficient use of physical-layer diversity
  - Opportunistic channel usage
  - Integrating application knowledge in network protocol design
- Security and Privacy
  - Physical-layer diversity to counter attackers
  - Distributed detection of misbehavior

Future Research: Multihop Wireless Networks

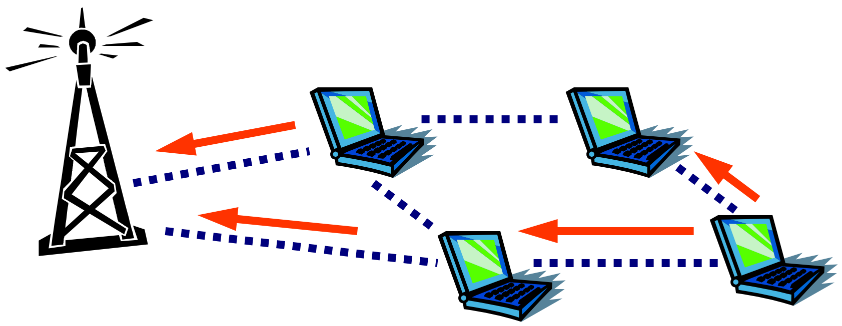
- Experimental testbeds
  - Test protocols in a realistic setting
  - Address implementation issues
  - Prior experience
    - Implementation in TinyOS on sensor hardware
    - User-level routing protocol for hybrid networks limited to several hops from access point



Implementation Experience

- Power save broadcast protocol in TinyOS on Mica2 motes

- User-level routing protocol for ad hoc networks limited to several hops from access point



Research Summary

- Secure Key Distribution [IEEE Infocom 2006]
- Adaptive Energy-Saving Protocols

	Adaptive Listening	**Adaptive Sleeping**
In-Band	[IEEE MASS 2005]	Multilevel routing [IEEE Broadnets 2004]
Out-of-Band	Our techniques are applicable	[IEEE WCNC 2004, IEEE Trans. on Mobile Computing 2005]

- Energy-Latency Tradeoff for Broadcast Dissemination [IEEE ICDCS 2005]
- Implementation Experience in TinyOS (sensors) and Linux

Thank You!

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

mjmille2@uiuc.edu

Acknowledgements to my adviser Prof. Nitin Vaidya, Prof. Indranil Gupta, Cigdem Sengul, and my research group

Sources (1/2)
(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

Sources (2/2)

- Champaign-Urbana Community Wireless Network (CUWiN)
  - http://cuwireless.net/
- DakNet
  - http://www.firstmilesolutions.com/products.php?p=daknet

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

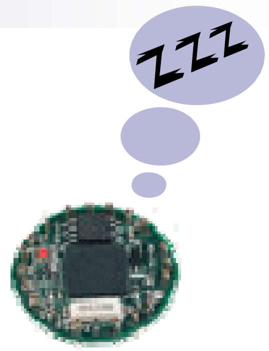
Is time synchronization a problem?

- 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

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?*

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

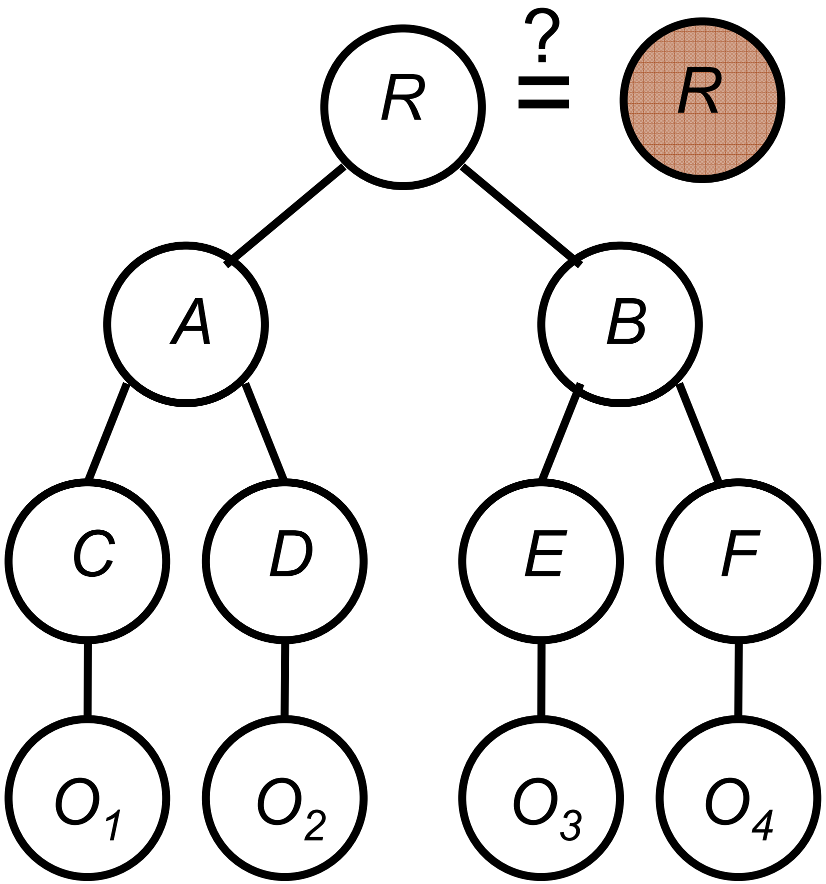
Adaptive Sleeping Results

- Simulated using *ns-2* and Poisson traffic
- Rate Estimation
  - Proposed protocol with Q=2.
- Optimal
  - Optimal value of S which minimizes energy over a single hop
- $S = \infty$
  - No timeout triggered wake-ups. Out-of-band wake-ups occur when Q=2 packets are in the queue.
- STEM
  - Out-of-band protocol proposed in [Schurgers02Optimizing]. Special case of our protocol with $S = \infty$ and Q=1.

Other Research

- Adaptive Framework for Energy-Saving Broadcast [IEEE ICDCS 2005]
  - Probabilistic protocol gives flexibility to choose tradeoffs in energy, latency, reliability, and overhead for broadcast dissemination
- Routing using multiple power save states
  - Metrics to find energy-efficient states for nodes on a path while achieving a desired latency

Merkle Tree Authentication



C = hash(O_1)

A = hash(C || D)

R = hash(A || B)

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