Improving Power Save Protocols Using Carrier Sensing for Dynamic Advertisement Windows

Matthew J. Miller
Nitin H. Vaidya
University of Illinois at Urbana-Champaign

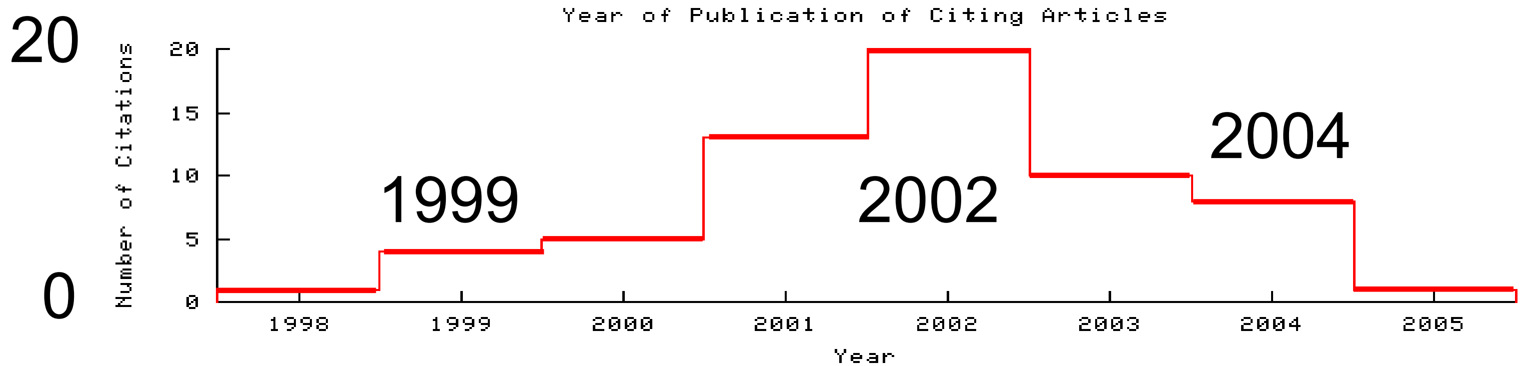
November 8, 2005

Lord of the Rings Protocol Design

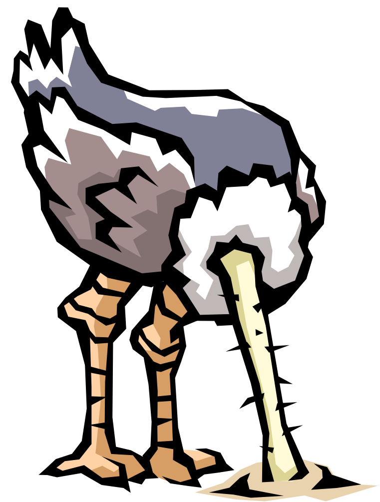
- Rohan sends Gondor a call for help
  - Sentries on hills miles apart light beacon fires in succession
- Given that events are infrequent, when a sentry wakes up and sees no beacons should they:
  A. Return to sleep immediately to save energy for other tasks
  B. Remain awake for a fixed amount of time just in case an event occurs during that period

Is Power Save Research Dead?

Citations for PAMAS paper by Year (from Citeseer)



- Unfortunately, a significant portion of sensor and ad hoc network research ignores the issue
  - Promiscuous listening
  - Frequent “Hello” messages
  - Latency of network-wide flooding

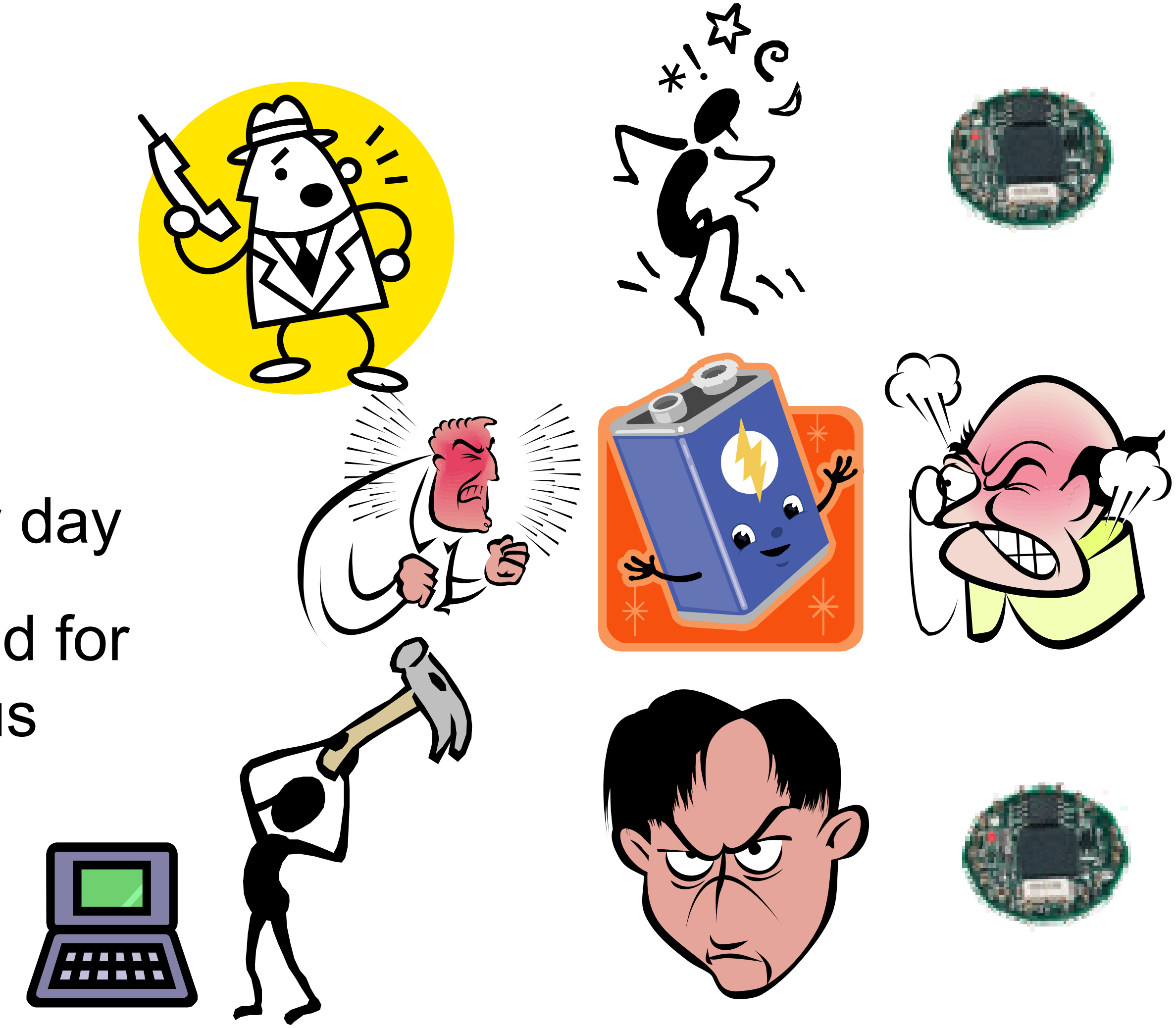


Is Power Save Research Important?

YES!!!

✓ It is a real world problem that affects wireless users every day

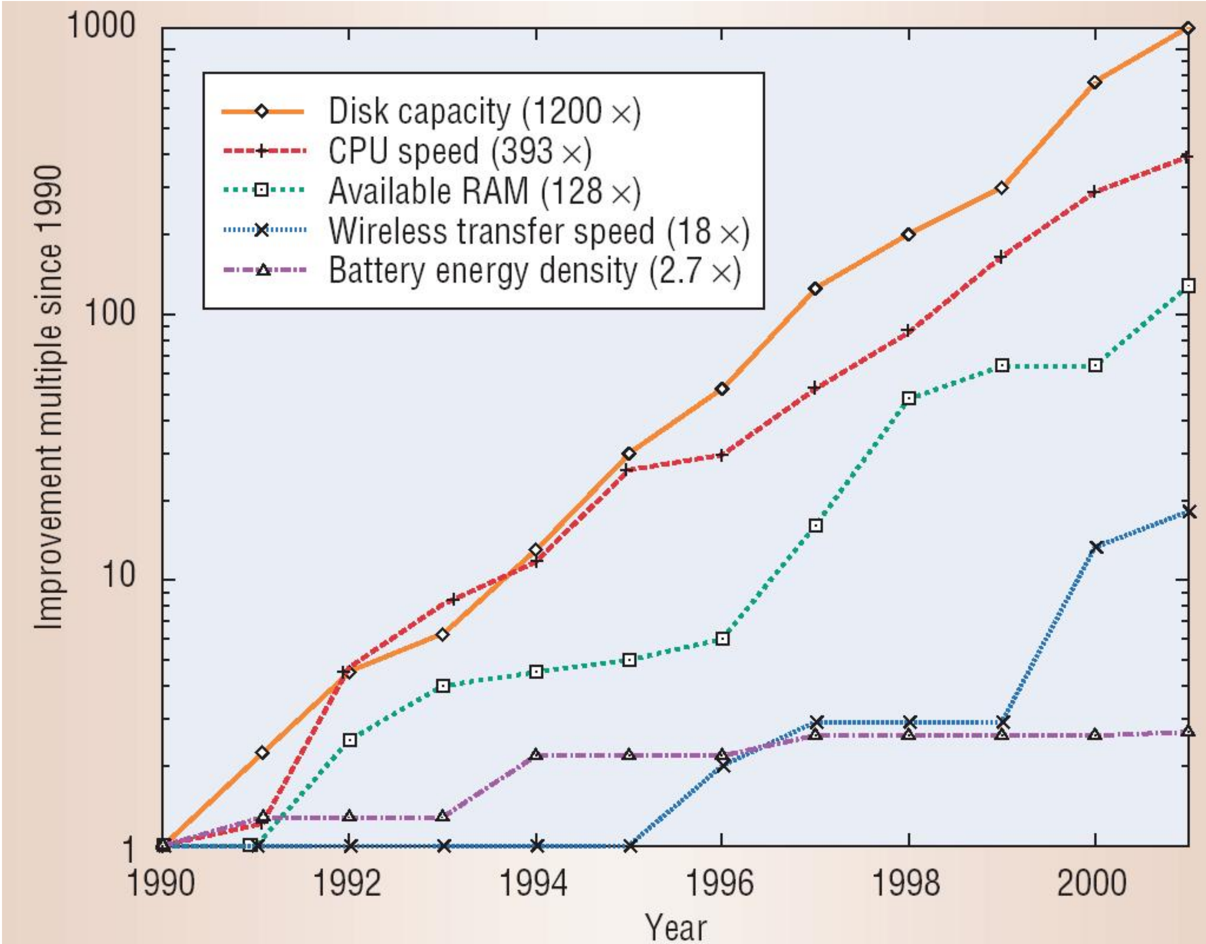
✓ Must be addressed for untethered ubiquitous wireless networks to become a reality



Won't Moore's Law Save Us?

NO!!!

Log Scale



1200 x

393 x

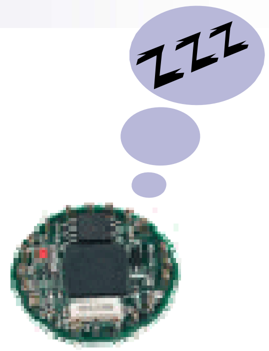
128 x

18 x

2.7 x

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

How to Save Energy at the Wireless Interface

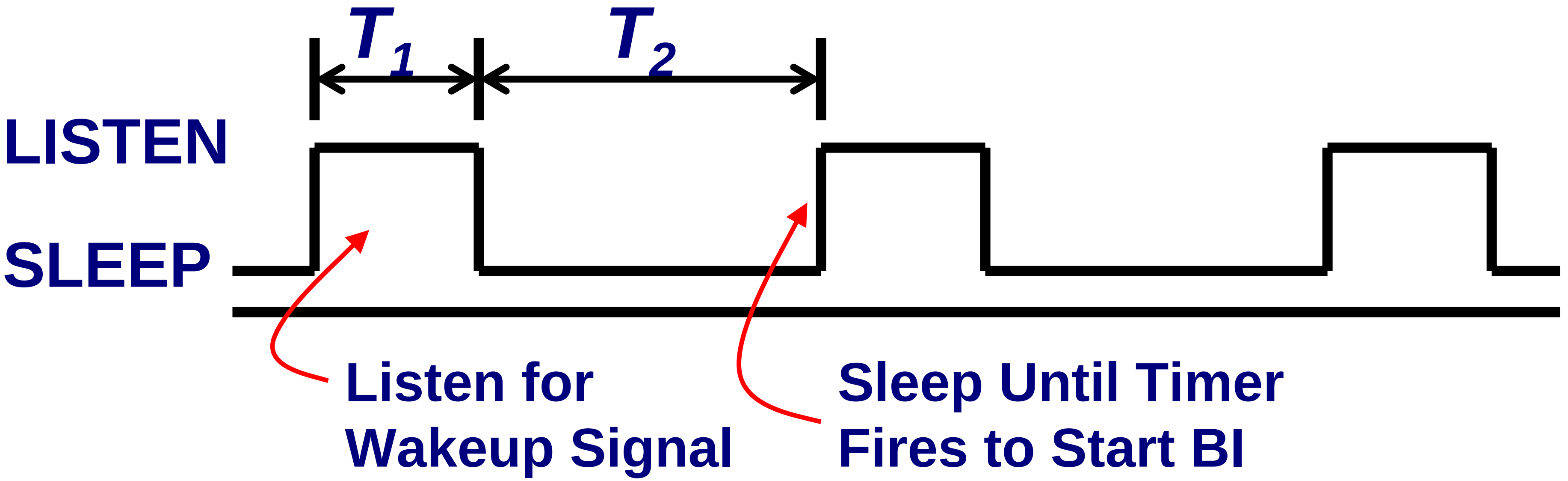


Specs for Mica2 Mote Radio

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

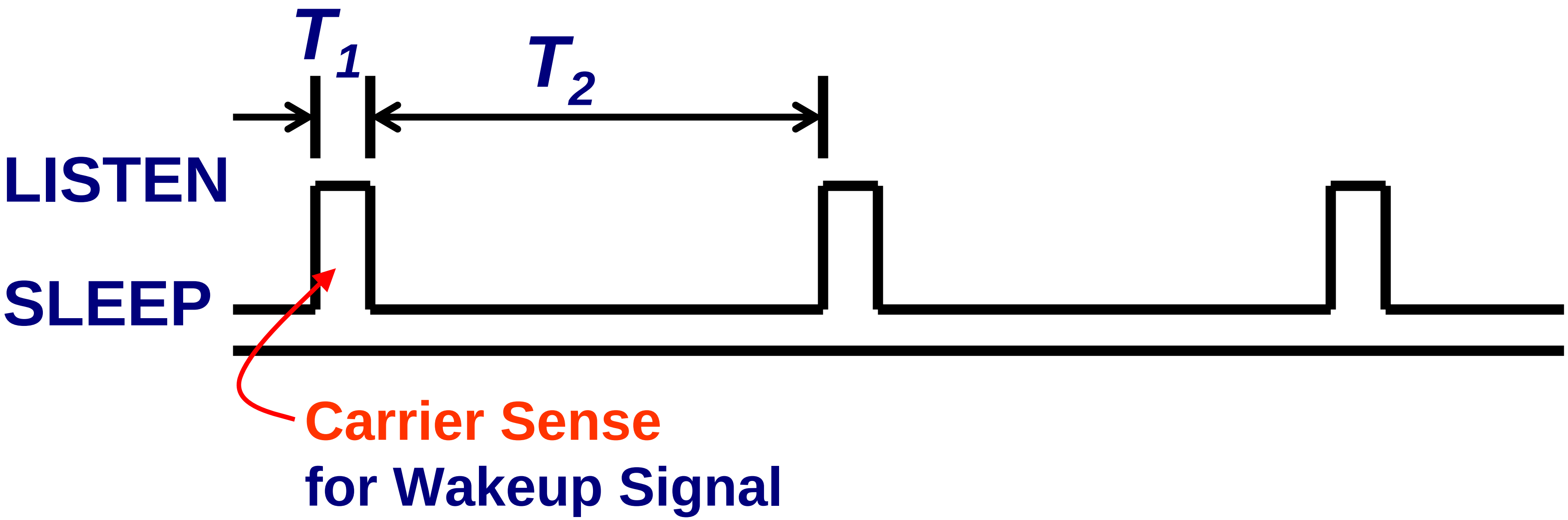
- Sleep as much as possible!!!
- Fundamental Question: *When should a radio switch to sleep mode and for how long?*
  - Many similarities in power save protocols since all are variations of these two design decisions

Common Design Used by Power Save Protocols



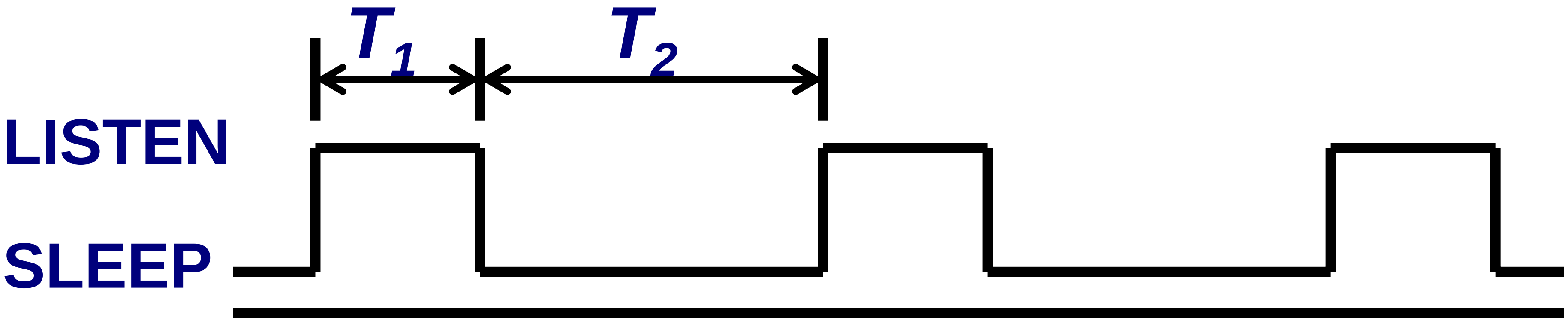
- $T_1 < T_2$
- Even with no traffic, node is awake for $T_1 / (T_1+T_2)$ fraction of the time
- T_1 is on the order of the time to receive a packet

Proposed Technique #1



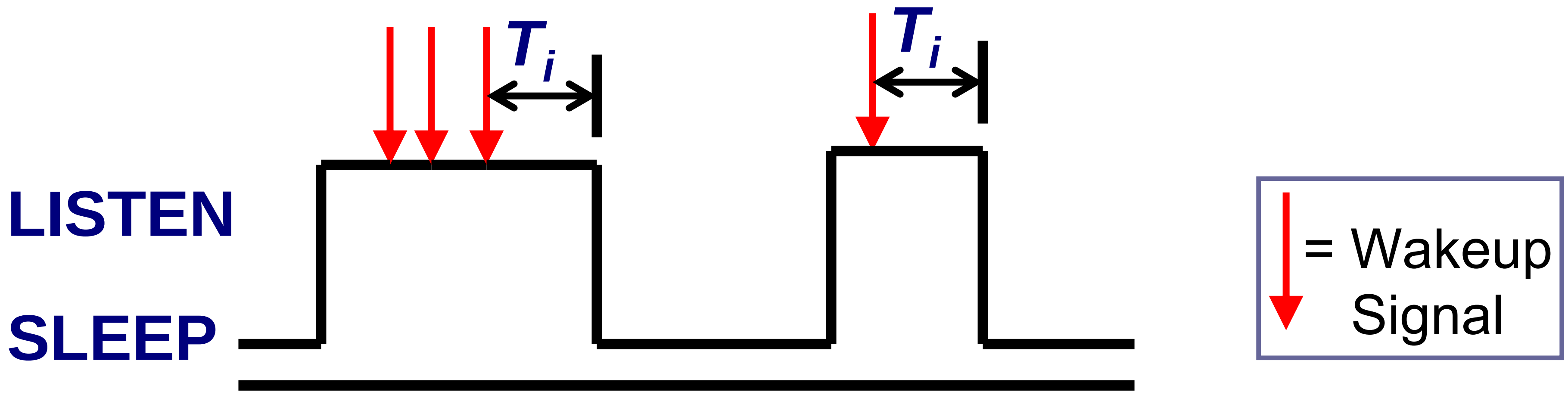
- Decrease T_1 using physical layer carrier sensing (CS)
- If carrier is sensed busy, then stay on to receive packet
- Typically, CS time << packet transmission time
  - E.g., 802.11 compliant hardware CS time ≤ 15 μs

Another Observation



- T_1 is fixed regardless of how many wakeup signals are received
- Ideally, nodes stay on just long enough to receive all wakeup signals sent by their neighbors
  - If no signals are for them → return to sleep

Proposed Technique #2

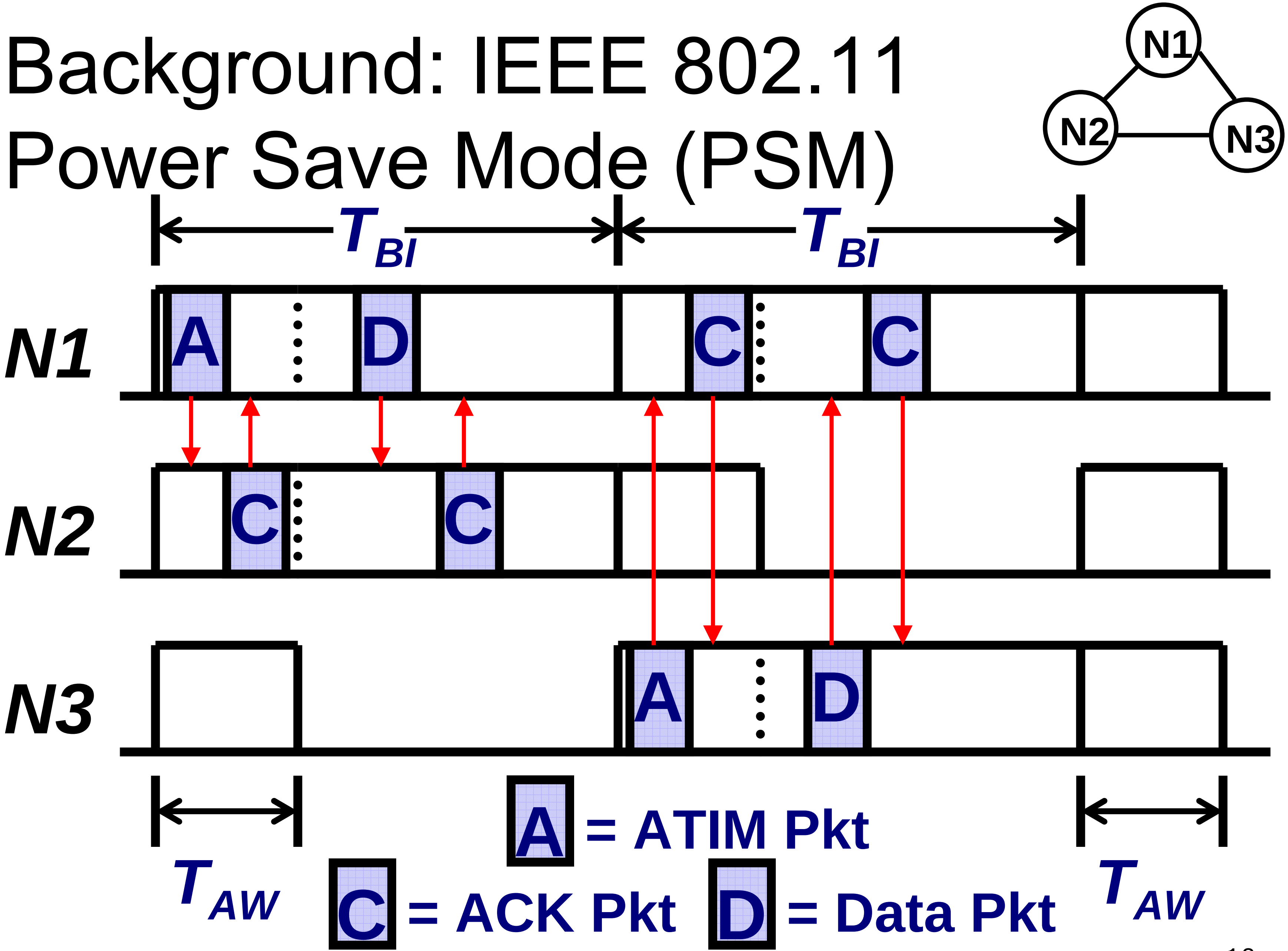


- Using physical layer CS, we dynamically extend the listening period for wakeup signals
- While previous work has proposed dynamic listening periods for 802.11 power save, ours is the first for single radio devices in multihop networks

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

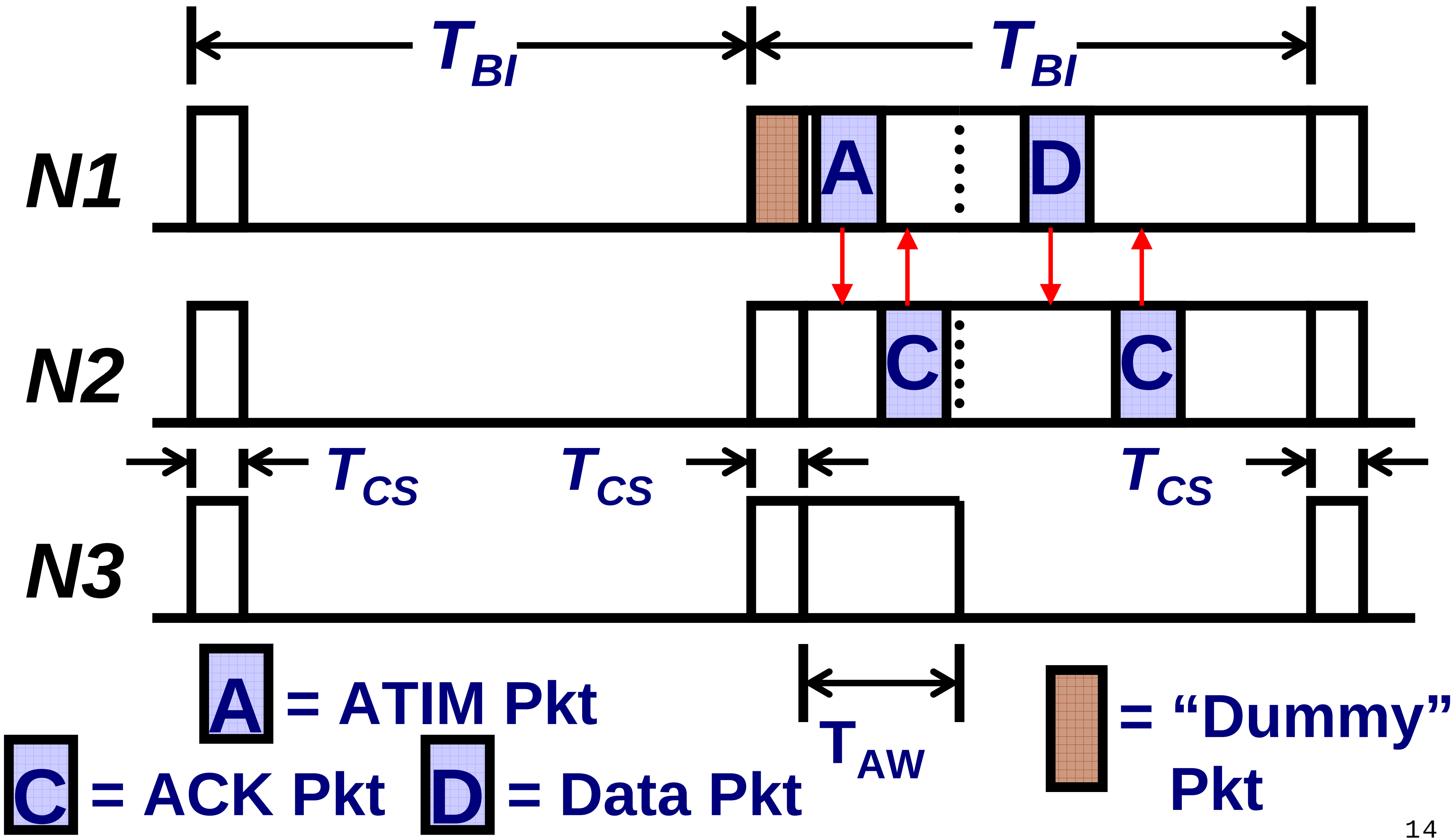
Background: IEEE 802.11 Power Save Mode (PSM)



Background: IEEE 802.11 PSM

- Nodes are assumed to be synchronized
  - In our protocols, we assume that time synchronization is decoupled from 802.11 PSM
- Every beacon interval (BI), all nodes wake up for an ATIM window (AW)
- During the AW, nodes advertise any traffic that they have queued
- After the AW, nodes remain active if they expect to send or receive data based on advertisements; otherwise nodes return to sleep until the next BI

Applying Technique #1 to 802.11 PSM



Applying Technique #1 to 802.11 PSM

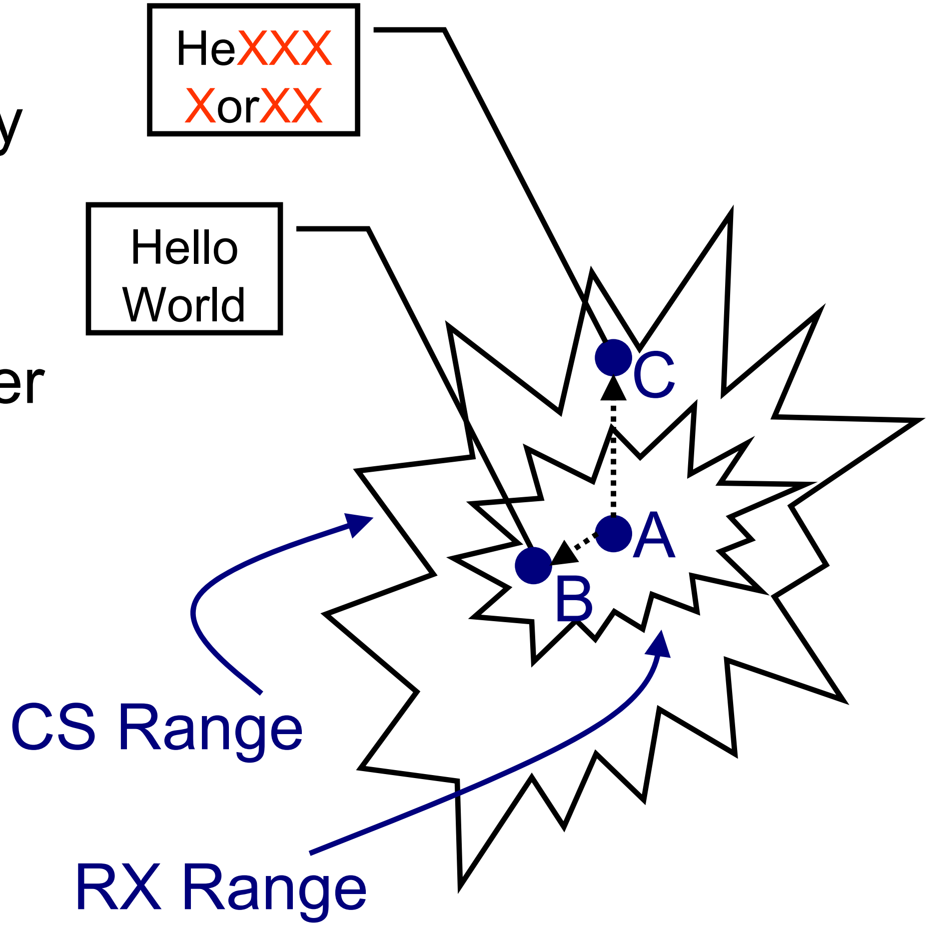
- Each beacon interval, nodes carrier sense the channel for T_{CS} time, where $T_{CS} \ll T_{AW}$
- If the channel is carrier sensed busy, nodes remain on for the remainder of the AW and follow the standard 802.11 PSM protocol
- If the channel is carrier sensed idle, nodes return to sleep without listening during the AW
- Node with data to send transmits a short "dummy" packet during T_{CS} to signal neighbors to remain on for AW

Observations

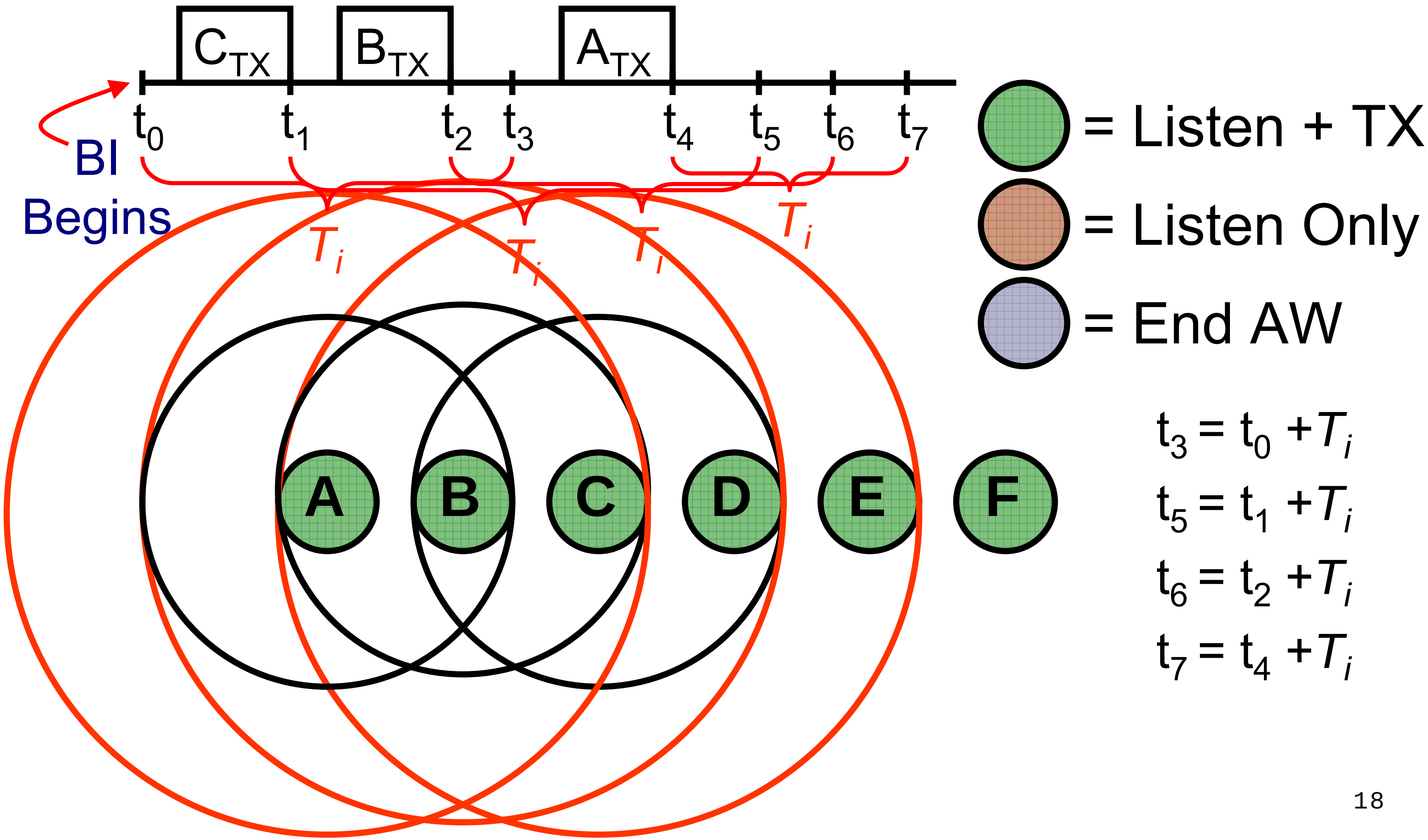
- When there are no packets to be advertised, nodes use significantly less energy
- Average latency is slightly longer
  - Packets that arrive during the AW are advertised in 802.11 PSM, but may not be with our technique
  - First packet cannot be sent until $T_{CS}+T_{AW}$ after beginning of BI instead of just T_{AW}
- 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)

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 AW scheme



Applying Technique #2 to 802.11 PSM



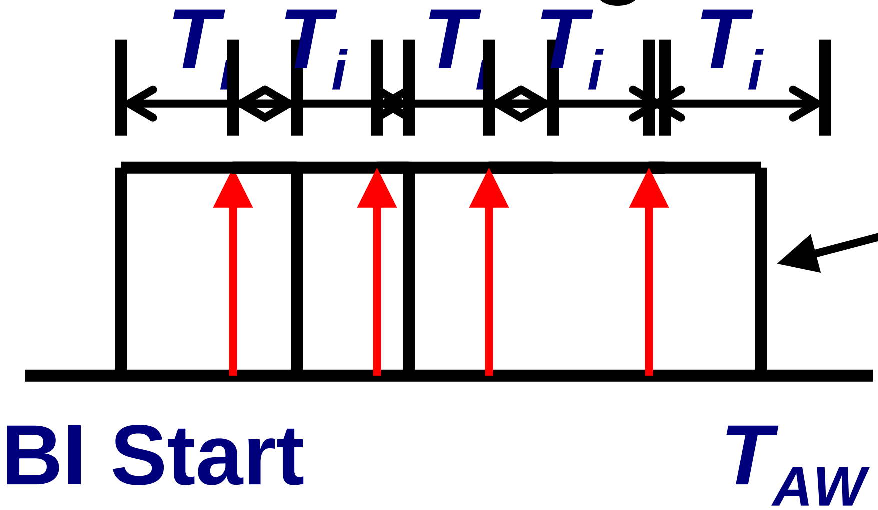
$t_3 = t_0 + T_i$

$t_5 = t_1 + T_i$

$t_6 = t_2 + T_i$

$t_7 = t_4 + T_i$

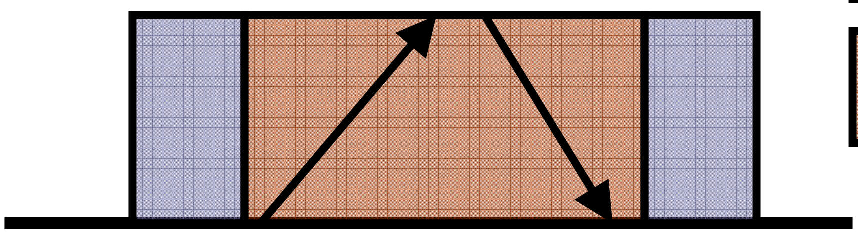
Applying Technique #2 to 802.11 PSM: Listening



Sleep according to 802.11 PSM rules

= TX, RX, or CS busy event

T_i =



= Max Contention Time

= ATIM/ATIM-ACK Handshake Time

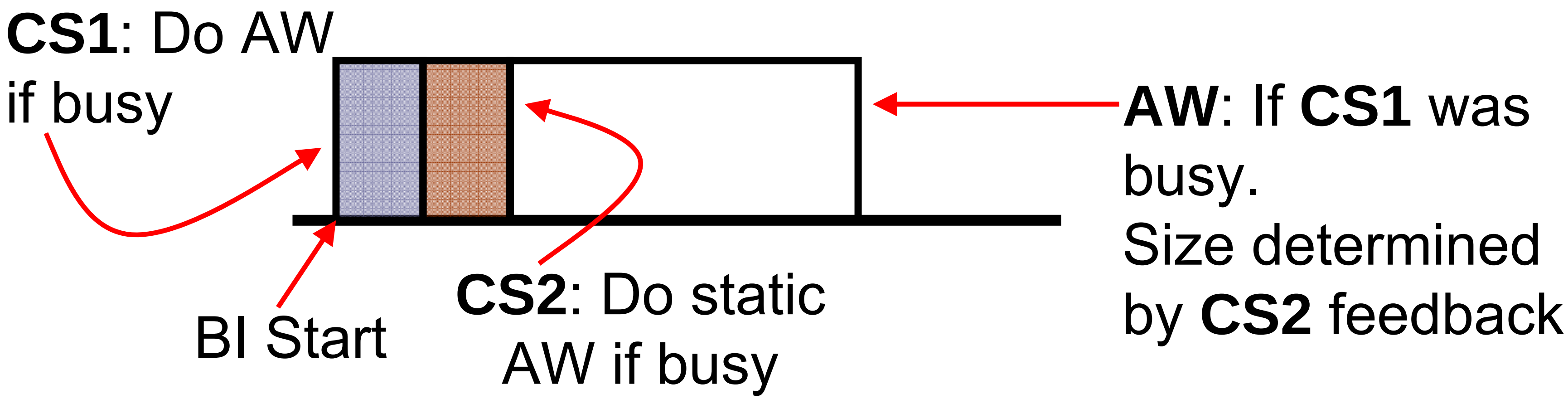
Applying Technique #2 to 802.11 PSM: Listening

- At the beginning of each BI, listen for T_i time ($T_{CS} < T_i < T_{AW}$)
- When a packet is sent or received OR the channel is carrier sensed busy, extend listening time by T_i
- Set maximum on how long the listening time can be extended since the beginning of the BI

Applying Technique #2 to 802.11 PSM: Sending

- Node with packets to advertise
  - If a packet has been received above the RX Threshold within T_i time, all neighbors are assumed to be listening
  - Otherwise, the node conservatively assumes that its intended receiver(s) is sleeping and waits until the next beacon interval to advertise the packet
- T_i is set such that a sender can lose one MAC contention and its receiver will continue listening

Combining Technique #1 and Technique #2

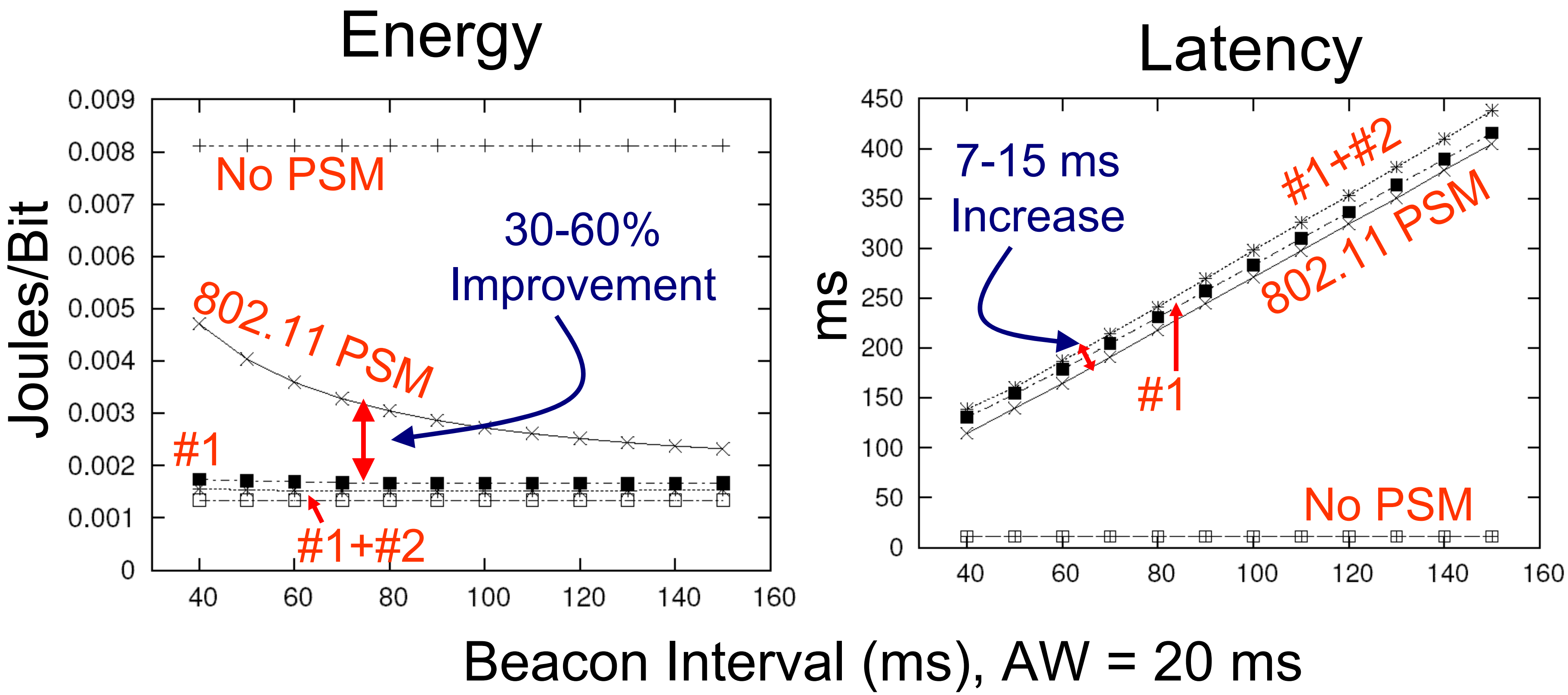


- First CS period indicates whether an AW is necessary
- Second CS period indicates whether AW size should be fixed or dynamic according to Technique #2
  - If a sender repeatedly fails using a dynamic AW, this is a fallback to the original protocol

ns-2 Simulation Setup

- 50 nodes placed uniformly at random in 1000 m × 1000 m area
- 2 Mbps radio with 250 m range
- Five flows with source and destination selected uniformly at random
  - Low traffic = 1 kbps per flow
  - Higher traffic = 10 kbps per flow
- Each data point averaged over 30 runs

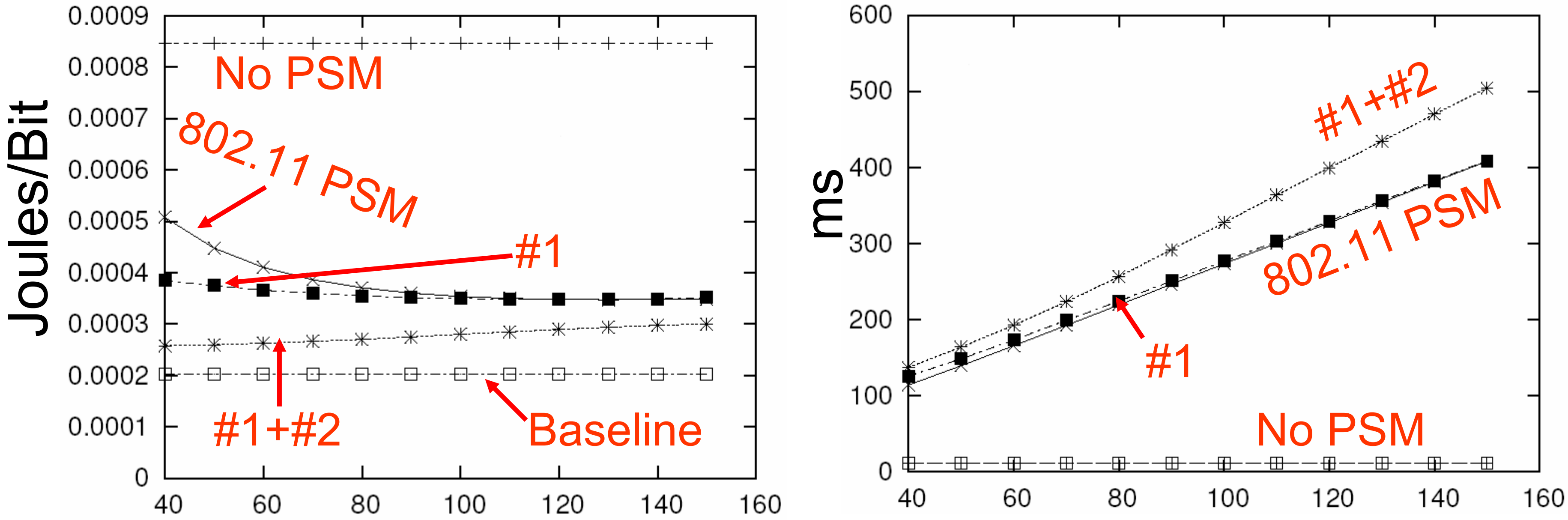
Low Traffic Results



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

Higher Traffic Results

Energy | Latency



Beacon Interval (ms), AW = 20 ms

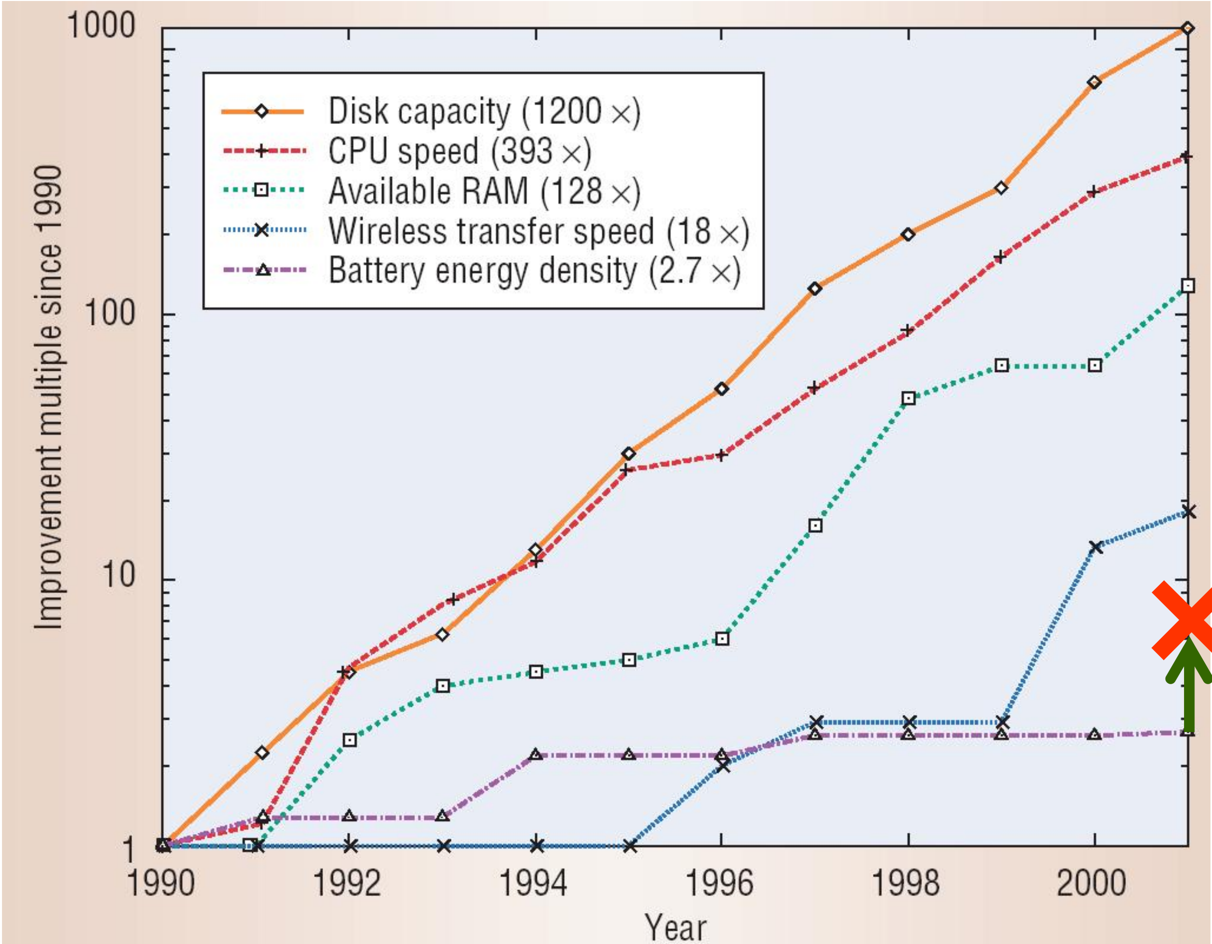
Differences from Lower Traffic: (1) More ATIM windows have at least one packet, (2) More contention means more deferred ATIMs

Conclusion

Every little bit helps…

But there's still a *long* ways to go!

Log Scale



1200 x

393 x

128 x

18 x

6.75 x

2.7 x

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

Thank You!!!

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

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

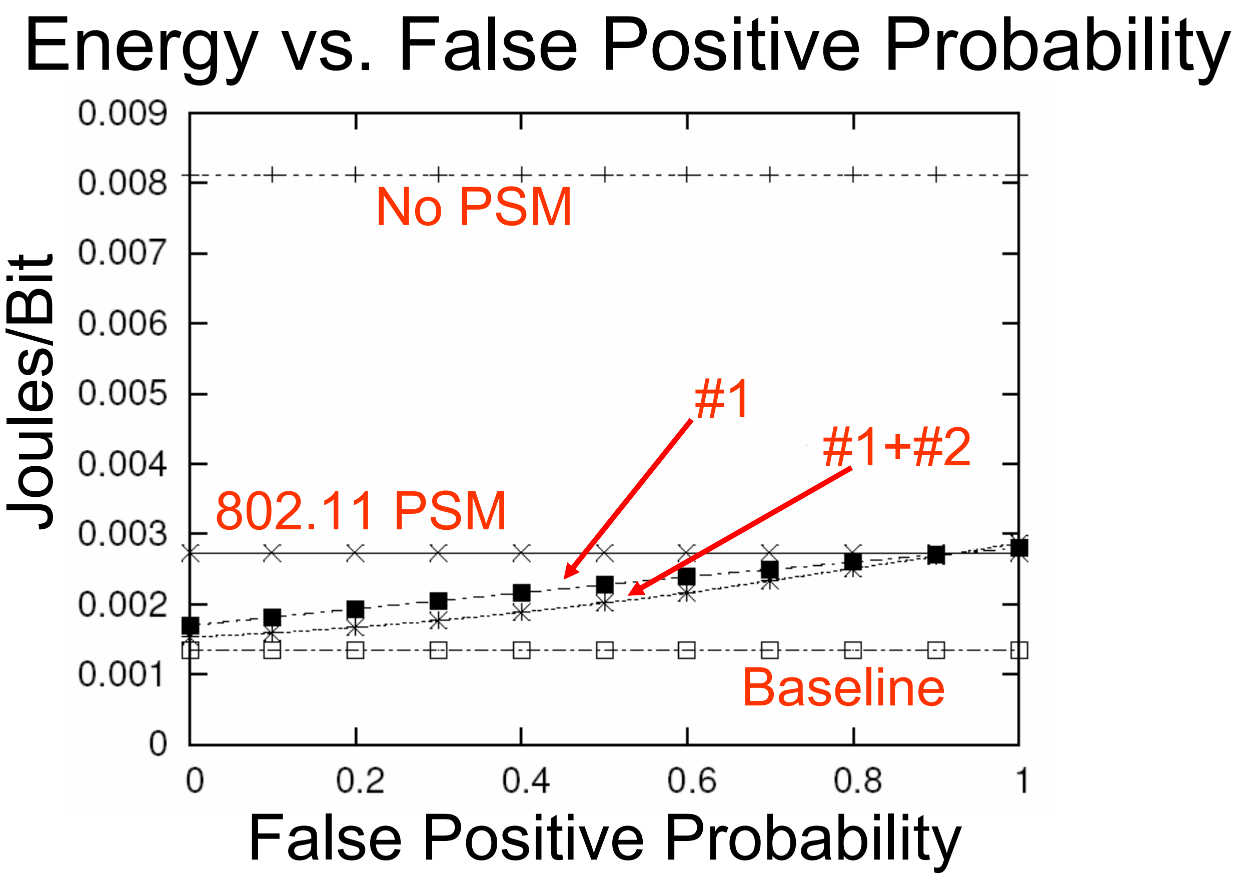
Is time synchronization a problem?

- 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	Wakeup 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

False Positive Results



Summary

- Less time spent checking and receiving wake-up signals and more time conserving energy
  - Application of physical layer CS to synchronous power save protocol to reduce listening interval
  - Physical layer CS for dynamic listening interval for single radio devices in multihop networks