

MAC protocols & research issues

- ❑ 802.11 wireless LAN (infrastructure mode)
 - MAC
 - research issues
- ❑ wireless sensor networks
 - MAC
 - research issues

Sensor nodes

□ A XSM sensor node (2004)

- 8MHz CPU, 4KB RAM, 128KB ROM
- Chipcon CC1000 radio: 19.2 kbps
- Infrared, acoustic, and magnetic sensors
- ...



□ Many more (2001 -)



courtesy: Hongwei Zhang, <http://www.cs.wayne.edu/~hzhang>

Wireless sensor networks



Science: ecology, seismology,
oceanography ...



Engineering: industrial automation,
precision agriculture, structural
monitoring ...



Daily life: traffic control,
health care, home security,
disaster recovery...

Wireless sensor networks

- ❑ Special ad hoc wireless network
- ❑ Large number of nodes w/ sensors & actuators
- ❑ Battery-powered nodes → energy efficiency
- ❑ Unplanned/planned deployment
- ❑ Node density & topology change → robustness
- ❑ Nodes cooperate for a common task
- ❑ In-network data processing

MAC in Sensor networks

- Important attributes of MAC protocols
 1. Collision avoidance
 2. Energy efficiency
 3. Scalability in node density
 4. Latency
 5. Fairness
 6. Throughput
 7. Bandwidth utilization

Many MAC protocols

- ❑ TDMA-based
- ❑ Random access
 - B-MAC ...
- ❑ Hybrid
 - S-MAC ...
- ❑ Adaptive to traffic
- ❑ Cross-layer design
 - Integrate MAC and higher-level protocols

outline

- S-MAC
- B-MAC
- Comparison of S-MAC & B-MAC
- Future directions

S-MAC (Sensor MAC)

- Design
 - Reduce energy consumption
- Main components
 - Periodic listen and sleep
 - Collision avoidance
 - Overhearing avoidance
 - Message passing

S-MAC design

- Major sources of energy waste
 - Idle listening ← Dominant in sensor nets
 - Long idle time when no sensing event happens
 - Collisions, control overhead, overhearing
- Combine benefits of TDMA & contention protocols
 - Periodic listen and sleep
 - Collision avoidance
 - Overhearing avoidance
 - Message passing

Periodic Listen and Sleep

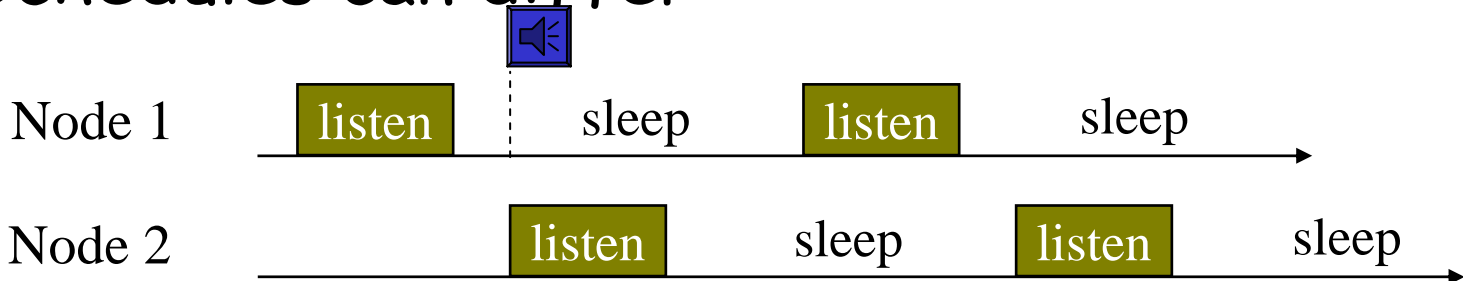
- ❑ Problem: Idle listening consumes significant energy
- ❑ Solution: Periodic listen and sleep



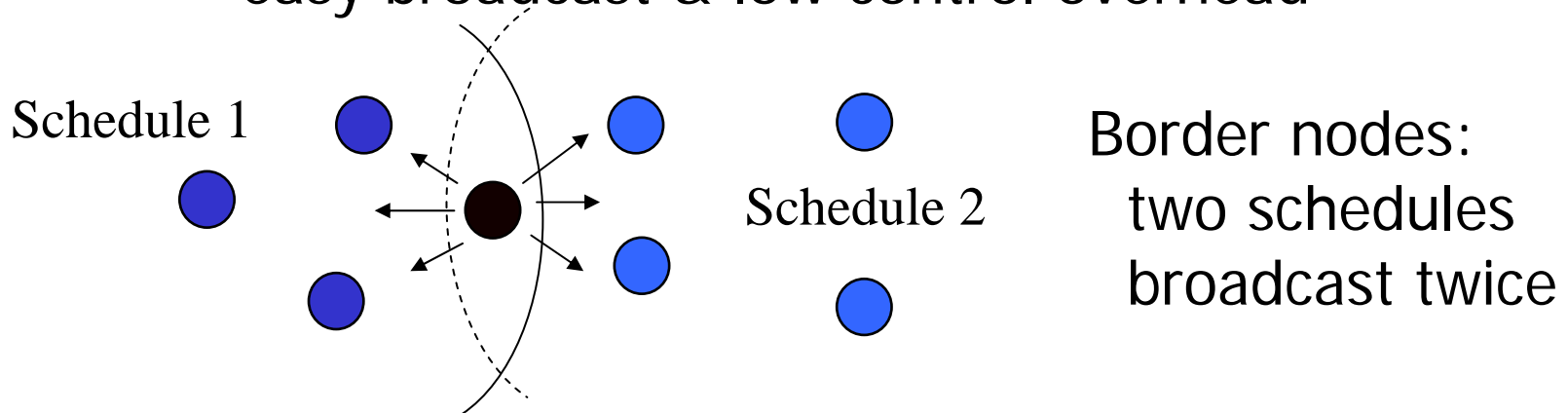
- Turn off radio when sleeping
- Reduce duty cycle to $\sim 10\%$ (200ms on/2s off)

Periodic Listen and Sleep

□ Schedules can differ



- *Prefer* neighboring nodes have same schedule
— easy broadcast & low control overhead



Periodic Listen and Sleep

□ Schedule Synchronization

- Remember neighbors' schedules
 - to know when to send to them
- Each node broadcasts its schedule every few periods of sleeping and listening
- Re-sync when receiving a schedule update
- Schedule packets also serve as beacons for new nodes to join a neighborhood

Collision Avoidance

- ❑ Problem: multiple senders want to talk
- ❑ Options: contention vs. TDMA
- ❑ Solution: similar to IEEE 802.11 ad hoc mode (DCF)
 - Physical and virtual carrier sense
 - Randomized backoff time
 - RTS/CTS for hidden terminal problem
 - RTS/CTS/DATA/ACK sequence

Overhearing Avoidance

- ❑ **Problem:** receive packets destined to others
- ❑ **Solution:** sleep when neighbors talk
 - Basic idea from PAMAS (Singh, Raghavendra 1998)
 - use in-channel signaling
- ❑ **Who should sleep?**
 - All immediate neighbors of sender & receiver
- ❑ **How long to sleep?**
 - The *duration* field in each packet informs other nodes the sleep interval

Message passing

- Problem: how to transmit *long* messages?
- Solution:
 - Long message is fragmented & sent in burst
 - RTS/CTS reserve medium for entire message
 - Fragment-level error recovery — ACK
 - extend Tx time and re-transmit immediately
- Other nodes sleep for whole message time

B-MAC (Berkeley MAC)

- ❑ Low power operation
- ❑ Effective collision avoidance
- ❑ Efficient channel utilization at low & high data rates
- ❑ Scalable to large numbers of nodes
- ❑ Simple implementation, small code & RAM size
- ❑ Reconfigurable by network protocols
- ❑ Tolerant to changing RF/Networking conditions

BMAC features

- ❑ Small, configurable
 - Export control to higher services
- ❑ Adaptive, reconfigurable
 - Adaptive bidirectional interface
 - Reconfigure based on the current workload
 - Identify best parameters at compile or run time (e.g., to maximize lifetime)

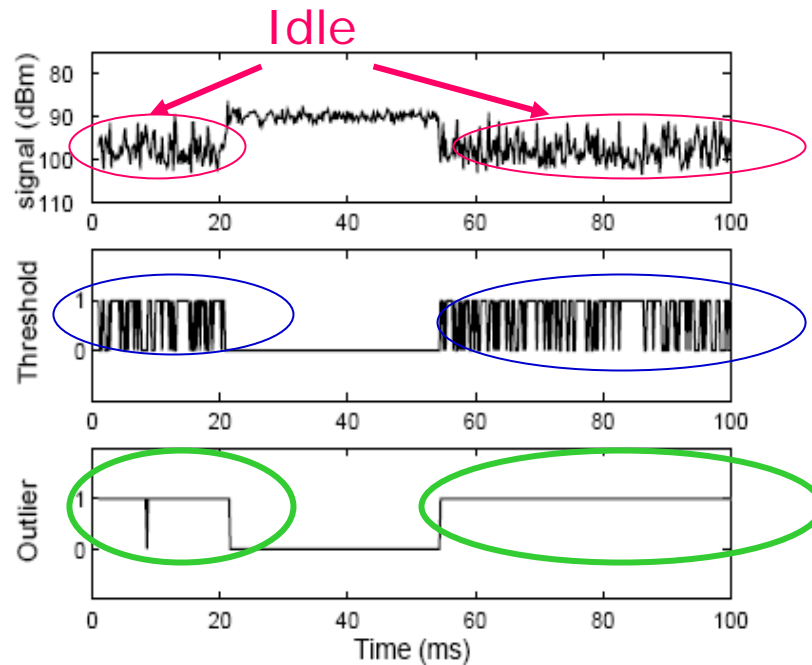
BMAC main components

- ❑ Only include core components
 - CSMA via CCA (Clear Channel Assessment) & backoff
 - Low power listening via Preamble
 - Optional ACK
- ❑ Enable/disable anything above

B-MAC Clear Channel Assessment

- Key observations
 - Ambient noise may change significantly depending on the environment
 - Packet reception has fairly constant channel energy
- Software approach to estimating noise floor
- Determine channel condition
 - Clear if find an outlier with energy much below noise floor, otherwise busy
 - Differ from threshold-based methods
 - Resilient to time-varying ambient noise

CCA vs. threshold techniques



CCA dynamically
adjusts threshold

CCA finds channel
busy/idle status with
high accuracy

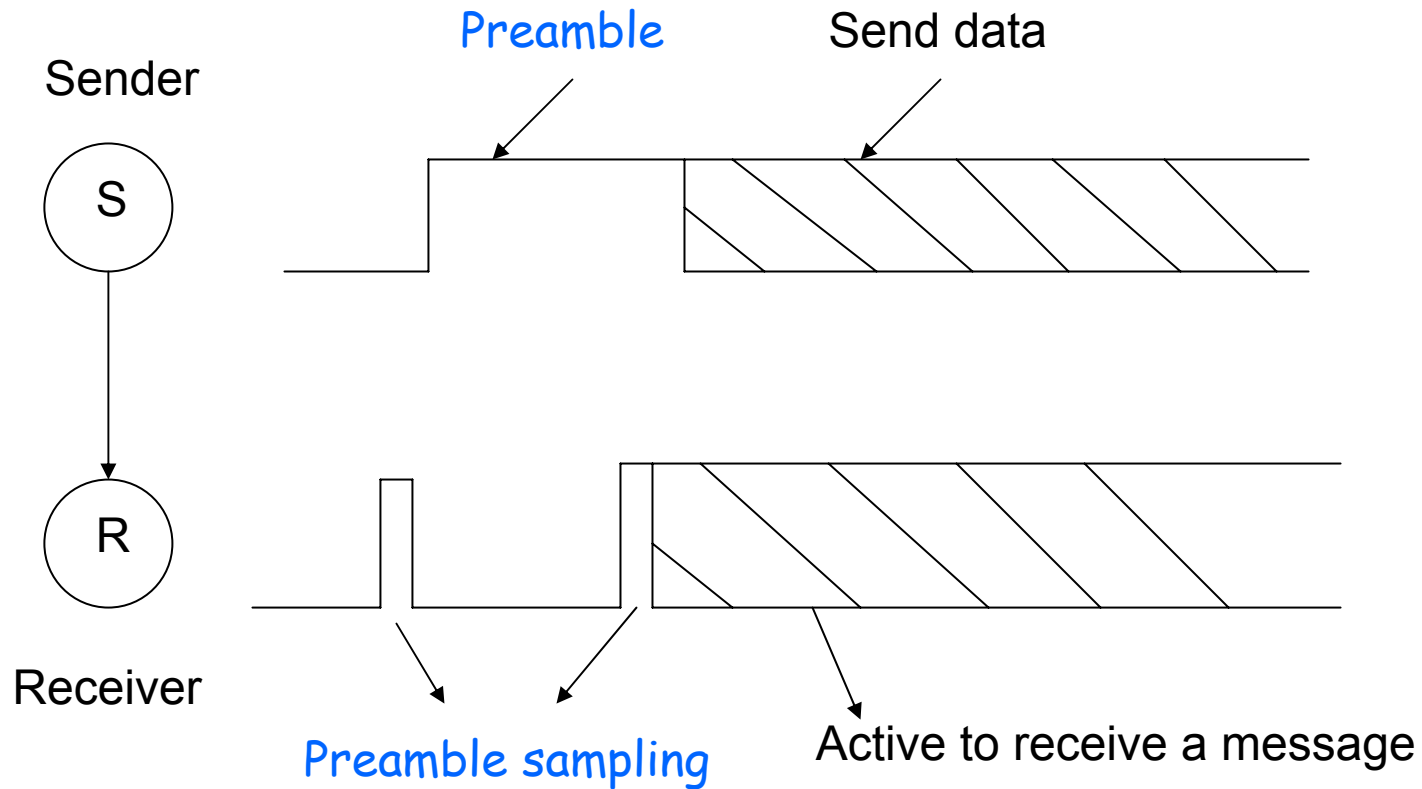
- Threshold: waste channel utilization
- CCA: Fully utilize the channel since a valid packet could have no outlier significantly below the noise floor

Using CCA

- ❑ CCA can be turned off
 - If off, schedule-based protocol (e.g., S-MAC) can be built on top B-MAC
- ❑ Using CCA
 - initial channel backoff when sending a message
 - Backoff length determined by upper-level service
 - Run outlier detection algorithm
 - If not clear, signal upper-level service to determine congestion backoff time

Low power listening

- Duty cycle radio through periodic channel sampling



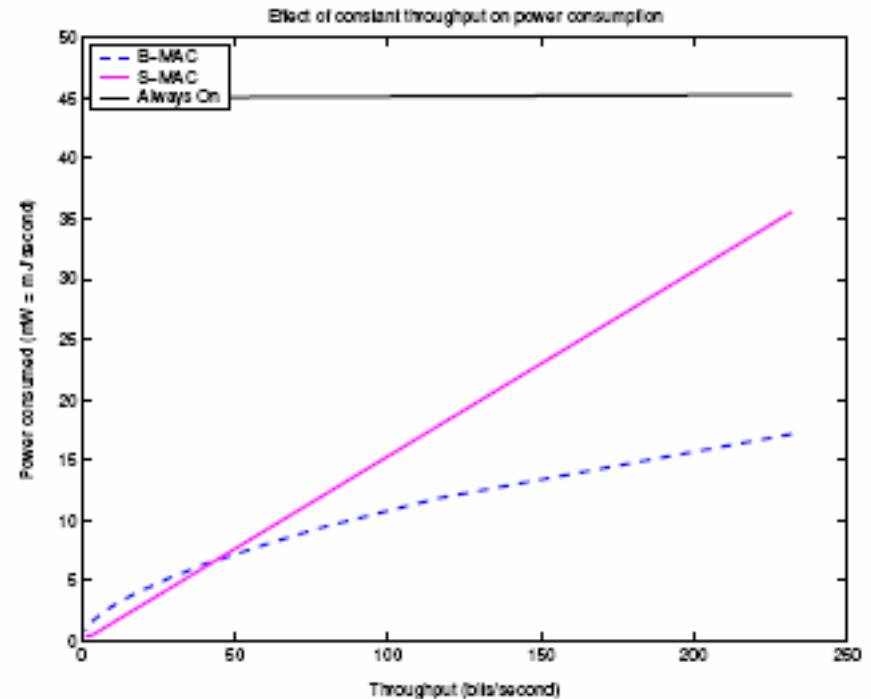
$$|\text{Preamble}| \geq \text{Sampling period}$$

S-MAC & B-MAC

- B-MAC is simpler than S-MAC
 - No RTS/CTS
 - You may have to implement RTS/CTS on your own...
 - No synchronization
 - No need for a schedule table in S-MAC
 - But periodic sleep & wake-up is a good approach to energy saving
- Experimental comparison in TinyOS
 - Set parameters in B-MAC to maximize lifetime

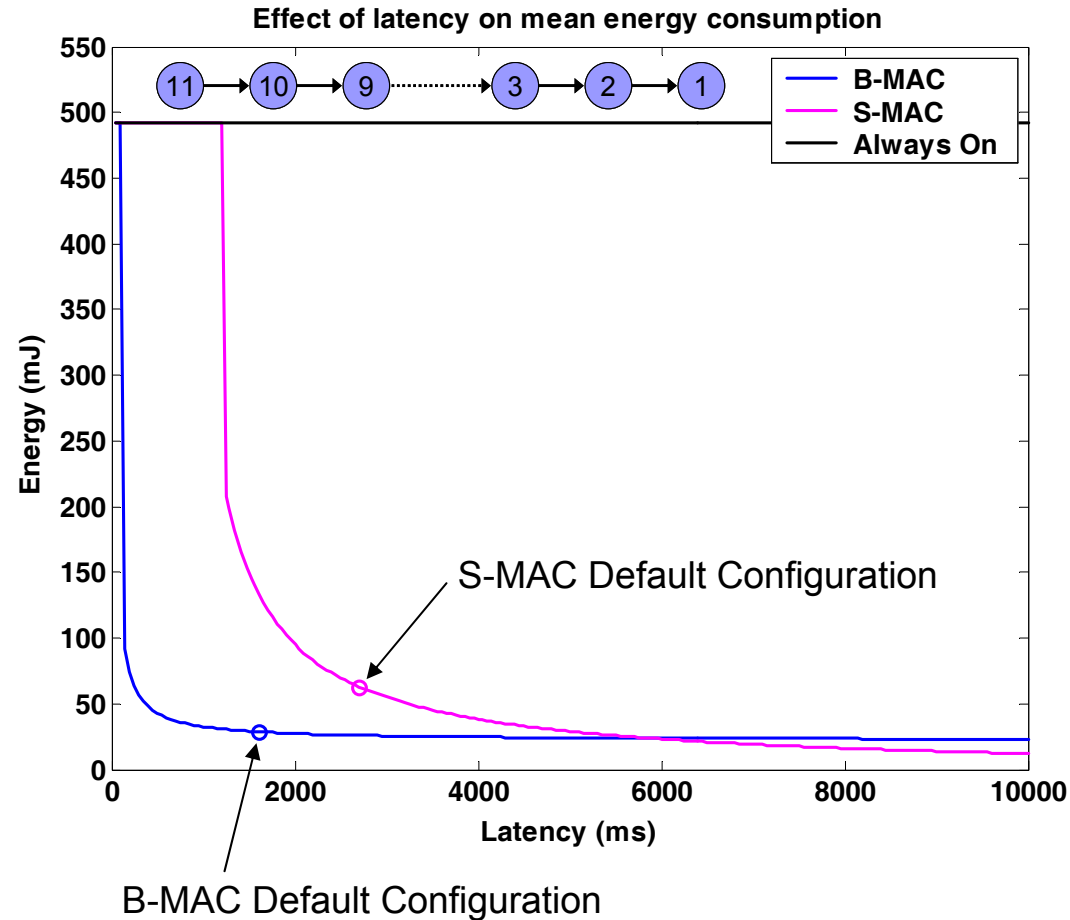
Throughput vs power consumption

- ❑ 10 nodes in a neighborhood
- ❑ Data must arrive within 10 seconds
- ❑ Average power consumption per node
- ❑ Low data rates
 - S-MAC is better
- ❑ high data rates
 - B-MAC is better



Energy vs Latency

- 10-hop network
- Source sends a 100-byte packet every 10 seconds



S-MAC & B-MAC summary

- B-MAC is
 - Easier to tune
 - better in channel assessment
 - not using explicit sync packets
 - not using RTS/CTS/ACK if it doesn't have to
 - smaller and less complex

Future directions

- ❑ Hard to find one-fit-all solution
 - Adaptive solutions
- ❑ Cross-layer design
 - Considering routing, MAC & scheduling together