Medium Access Control in Wireless Sensor Networks

Outline

- Introduction to MAC
- MAC attributes and trade-offs
- Scheduled MAC protocols
- Contention-based MAC protocols
- Case studies
- Summary

Introduction to MAC

- The role of medium access control (MAC)
  - Controls when and how each node can transmit in the wireless channel
- Why do we need MAC?
  - Wireless channel is a shared medium
  - Radios transmitting in the same frequency band interfere with each other – collisions
  - Other shared medium examples: Ethernet

Where Is the MAC?

- Network model from Internet

Application layer	
Transport layer	→ End-to-end reliability, congestion control
Network layer	→ Routing
Link/MAC layer	→ Per-hop reliability, flow control, multiple access
Physical layer	→ Packet transmission and reception

- A sublayer of the Link layer
  - Directly controls the radio
  - The MAC on each node only cares about its neighborhood

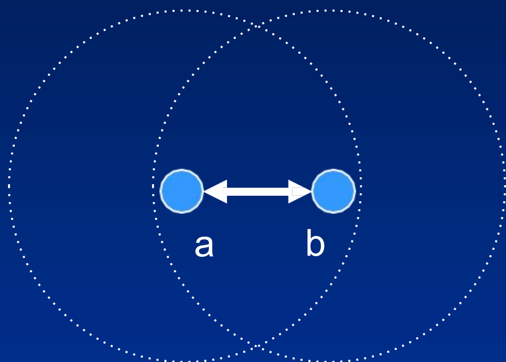
What's New in Sensor Networks?

- A special wireless ad hoc network
  - Large number of nodes
  - Battery powered
  - Topology and density change
  - Nodes for a common task
  - In-network data processing
- Sensor-net applications
  - Sensor-triggered bursty traffic
  - Can often tolerate some delay
    - Speed of a moving object places a bound on network reaction time

Next…

- Introduction to MAC
- MAC attributes and trade-offs
- Scheduled MAC protocols
- Contention-based MAC protocols
- Case studies
- Summary

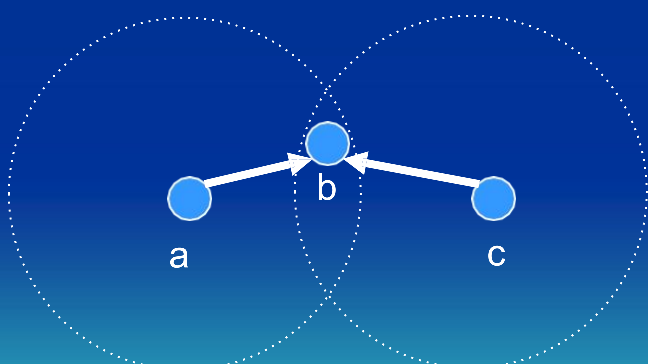
Interference / Collisions



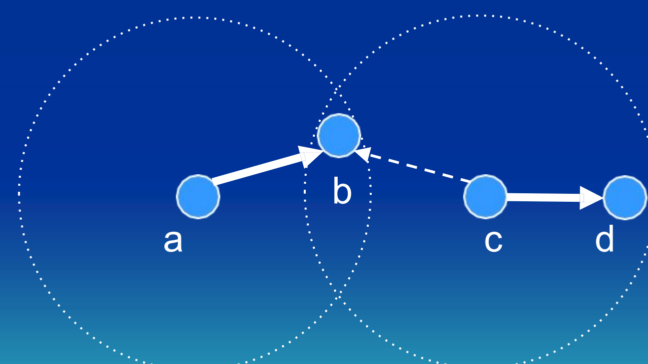
a and ***b*** interfere and hear noise only

Packets which suffered collisions should be re-sent.

Ideally, we would want all packets to be sent collision-free, only once…



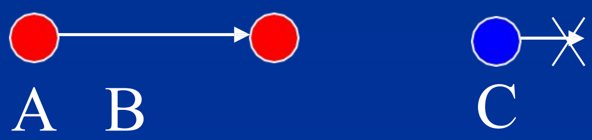
Interference on node ***b***
("Hidden terminal problem")



Interference on node ***b***

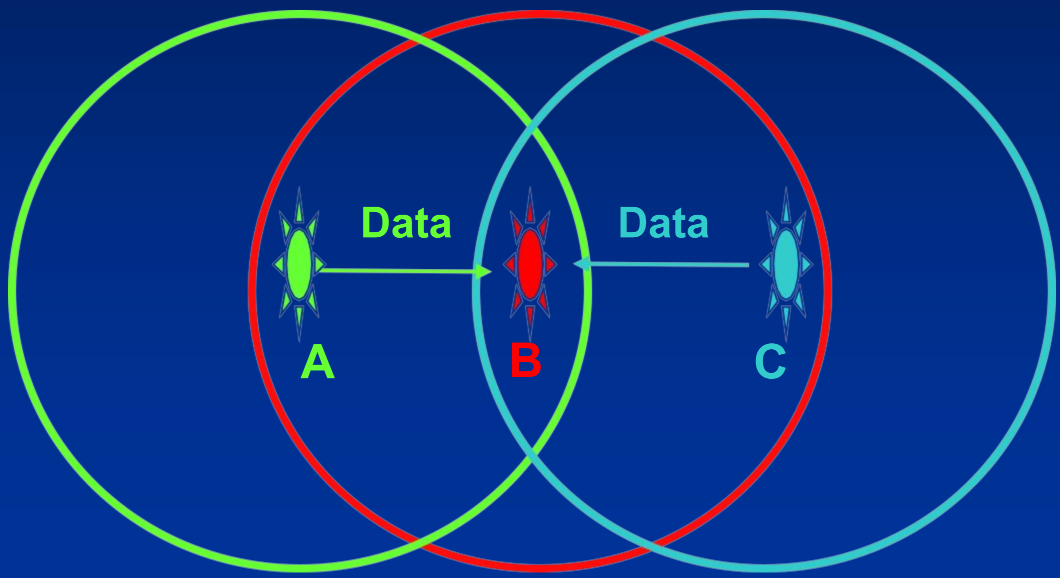
MACA Protocol

- Contention-based protocols
  - CSMA — Carrier Sense Multiple Access
    - Ethernet (CSMA/CD) is not enough for wireless (collision at receiver cannot detect at sender)



Hidden terminal: A is hidden from C's CS

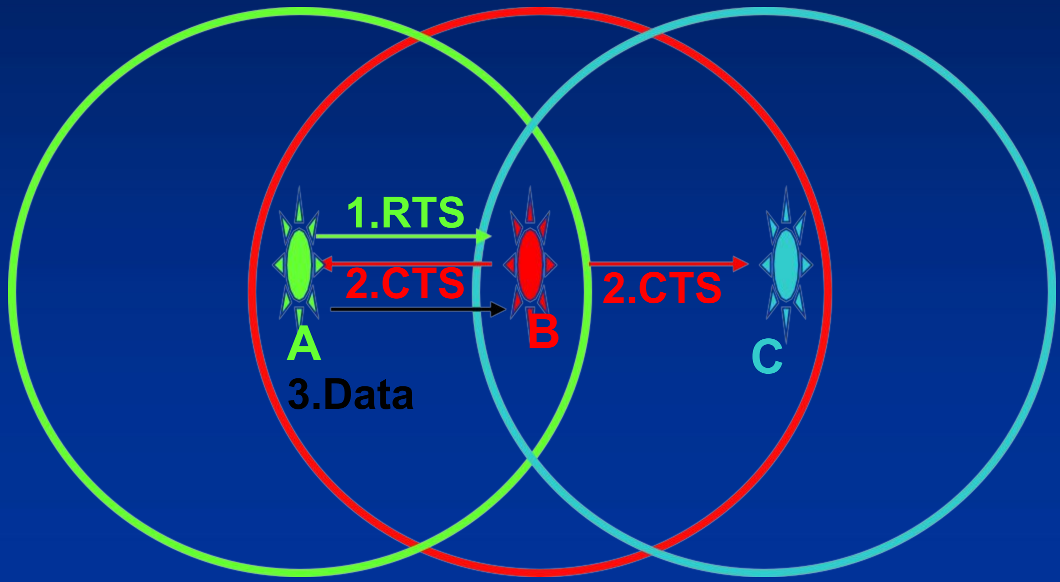
Hidden Terminal Problem



A and C want to send data to B

1. A senses medium idle and sends data
2. C senses medium idle and sends data
3. Collision occurs at B

Collision Avoidance w/ RTS/CTS



A and C want to send to B

1. A sends RTS (Request To Send) to B
2. B sends CTS (Clear To Send) to A
   C "overhears" CTS from B
3. C waits for duration of A's transmission

Next…

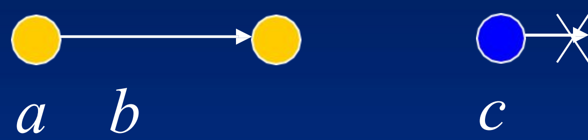
- Introduction to MAC
- MAC attributes and trade-offs
- Scheduled MAC protocols
- Contention-based MAC protocols
- Case studies
- Summary

Contention Protocols: Classics

- ALOHA
  - Pure ALOHA: send when there is data
  - Slotted ALOHA: send on next available slot
  - Both rely on retransmission when there’s collision
- CSMA — Carrier Sense Multiple Access
  - Listening (carrier sense) before transmitting
  - Send immediately if channel is idle
  - Backoff if channel is busy
    - non-persistent, 1-persistent and p-persistent

Contention Protocols: CSMA/CA

- Hidden terminal problem



Node *a* is hidden from *c*'s carrier sense

  - CSMA is not enough for multi-hop networks (collision at receiver)
- CSMA/CA (CSMA with Collision Avoidance)
  - RTS/CTS handshake before send data
  - Node *c* will backoff when it hears *b*'s CTS

Contention Protocols: IEEE 802.11

- IEEE 802.11 ad hoc mode (DCF)
  - Virtual and physical carrier sense (CS)
    - Network allocation vector (NAV), duration field
  - Binary exponential backoff
  - RTS/CTS/DATA/ACK for unicast packets
  - Broadcast packets are directly sent after CS
  - Fragmentation support
    - RTS/CTS reserve time for first (fragment + ACK)
    - First (fragment + ACK) reserve time for second…
    - Give up transmission when error happens

Contention Protocols: IEEE 802.11 (cont.)

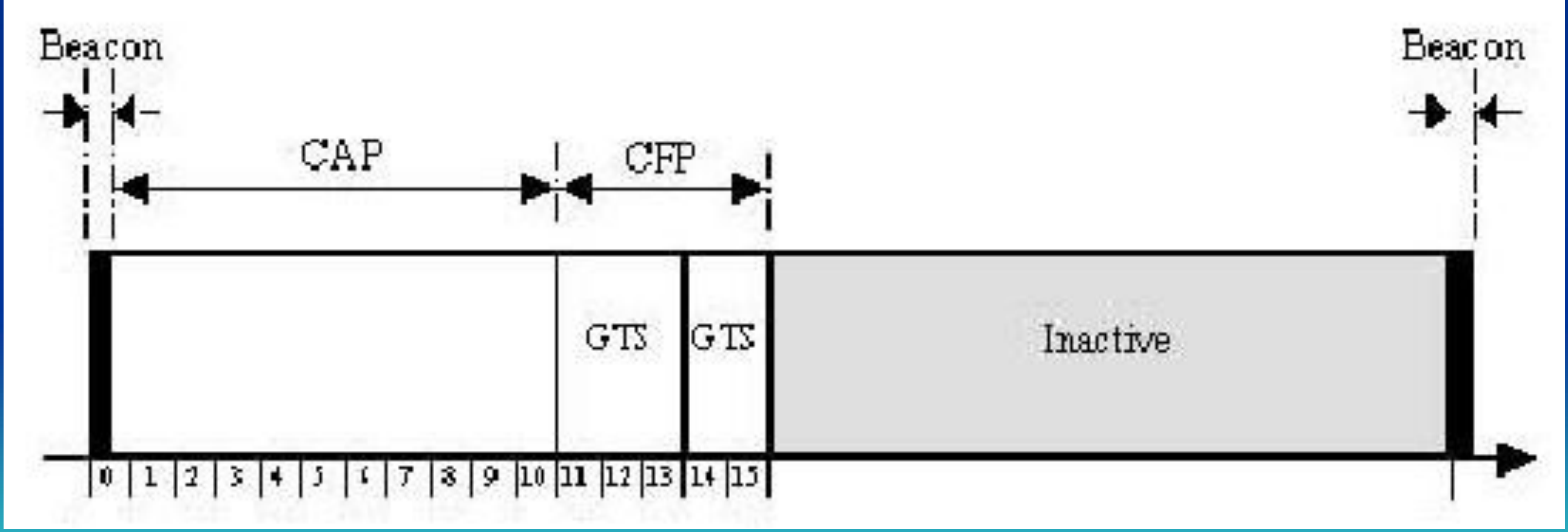
- Power save (PS) mode in IEEE 802.11 DCF
  - Assumption: all nodes are synchronized and can hear each other (single hop)
  - Nodes in PS mode periodically listen for beacons & ATIMs (ad hoc traffic indication messages)
  - Beacon: timing and physical layer parameters
    - All nodes participate in periodic beacon generation
  - ATIM: tell nodes in PS mode to stay awake for Rx
    - ATIM follows a beacon sent/received
    - Unicast ATIM needs acknowledgement
    - Broadcast ATIM wakes up all nodes — no ACK

Contention Protocols: ZigBee

- Based on IEEE 802.15.4 MAC and PHY
  - Three types devices
    - Network Coordinator
    - Full Function Device (FFD)
      - Can talk to any device, more computing power
    - Reduced Function Device (RFD)
      - Can only talk to a FFD, simple for energy conservation
  - CSMA/CA with optional ACKs on data packets
  - Optional beacons with superframes
  - Optional guaranteed time slots (GTS), which supports contention-free access

Contention Protocols: ZigBee (cont.)

- Low power, low rate (250kbps) radio
- MAC layer supports low duty cycle operation
  - Target node life time > 1 year



Next…

- Introduction to MAC
- MAC attributes and trade-offs
- Scheduled MAC protocols
- Contention-based MAC protocols
- Case studies
- Summary

MAC Protocols for WSNs

- Asynchronous MAC protocols
  - No synchronization or coordinate schedule between neighbor nodes
  - S-MAC, T-MAC, B-MAC, Wise MAC, etc.
- Synchronous MAC protocols
  - Time synchronization is achieved externally or synchronization is managed by specific node
  - TRAMA, DMAC, LEACH, etc.

S-MAC

- S-MAC assume sensor networks to be composed of many small nodes deployed in an ad hoc fashion.
  - The large number of nodes can also take advantage of short-range, multi-hop communication to conserve energy.
  - Most communication will be between nodes as peers, rather than to a single base-station.

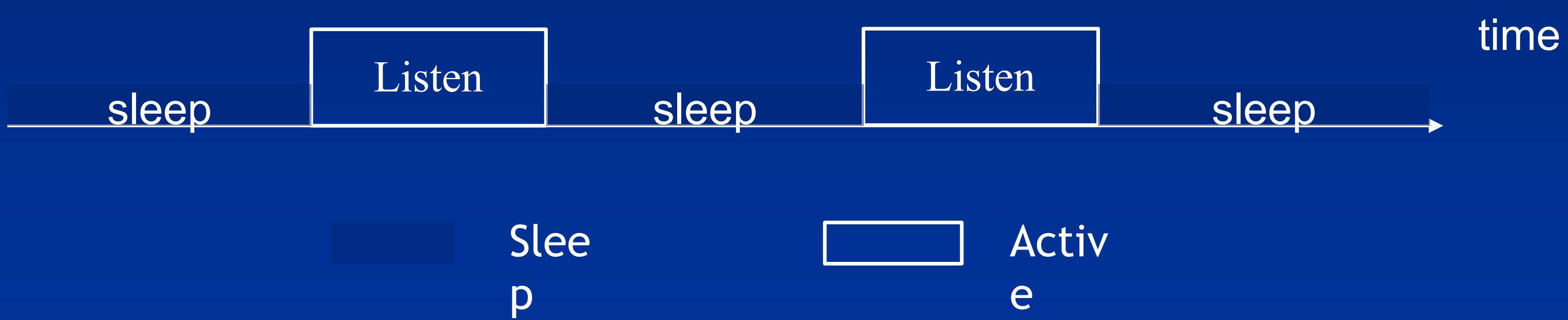
S-MAC

- S-MAC designed for reduce energy consumption and support self-configuration
  - To reduce energy consumption in listening to an idle channel, nodes *periodically sleep*
  - Neighboring nodes form *virtual clusters* to auto-synchronize on sleep schedules
  - S-MAC applies *message passing* to reduce contention latency for sensor-network applications

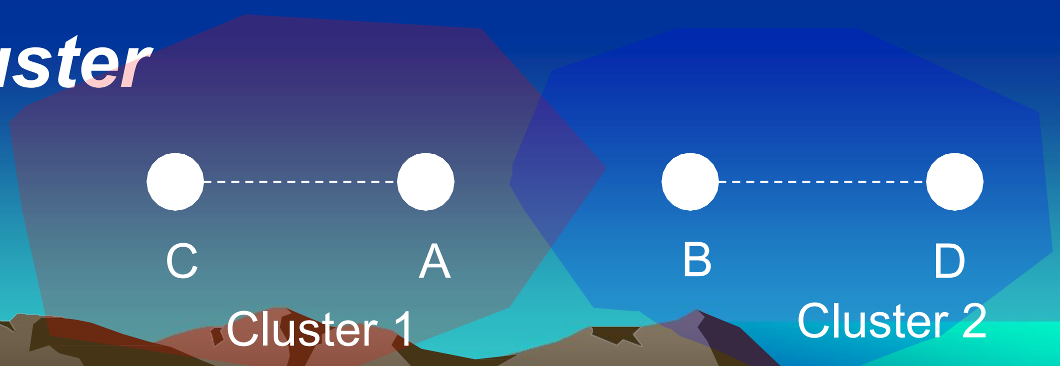
  - https://www.youtube.com/watch?v=FmE1b7ETVd0

S-MAC

- **Locally managed synchronizations periodic sleep–listen schedules**



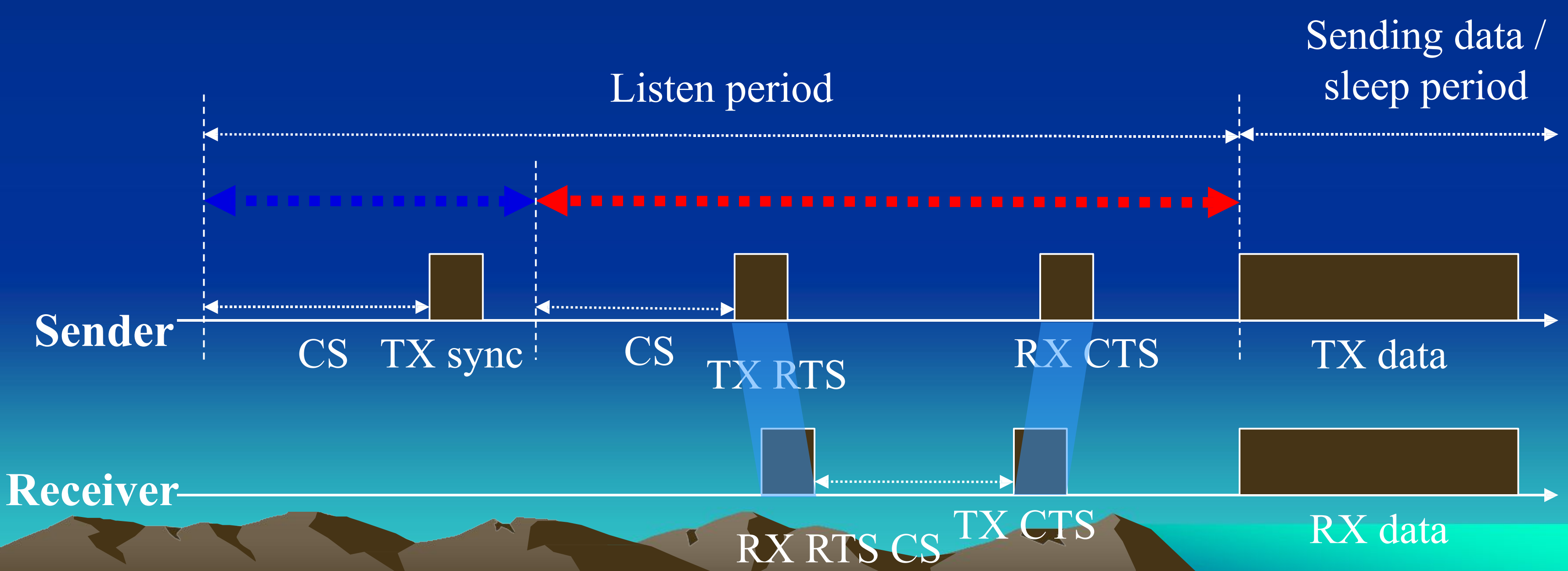
- ***Virtual cluster***



S-MAC

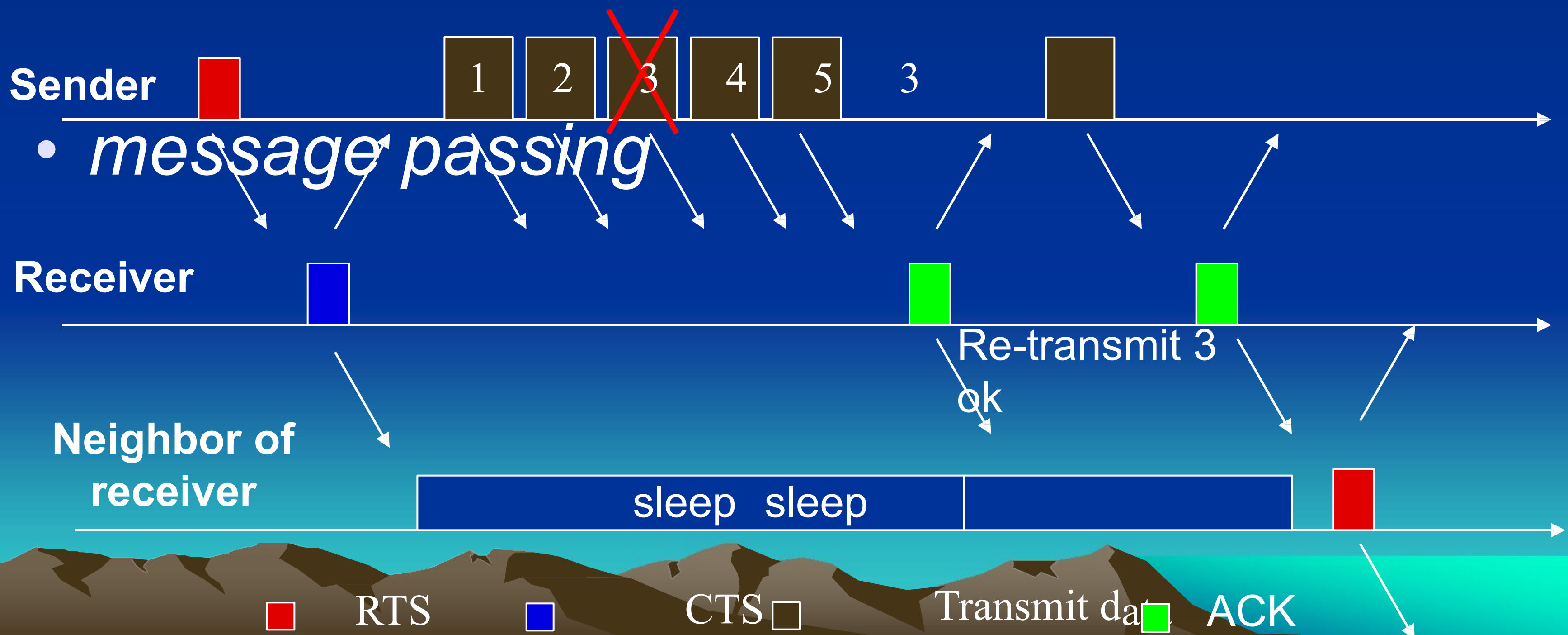
- Every node should wakeup in Listen period
  - Synchronization period
  - Control period (RTS/CTS)

※ Node use CSMA before sending any packet



S-MAC

- **Re-transmit message problem**
  - **Long message => re-transmission will take a long time**
  - **Short message => large control overhead (RTS/CTS)**
- *message passing*



S-MAC

- Adaptive-Listening
  - Node who overhears its neighbor’s transmissions (ideally only RTS or CTS) wake up for a short period of time at the end of the data transmission.

•If the node is the next-hop node => remain active after data transmission, prepare to forwarding its neighbor’s message.

•If the node does not receive anything during the adaptive listening => go back to sleep.

S-MAC-Summary

- Locally time synchronization between neighbors
- Power saving method:
  - Fixed wakeup/sleep interval
- Transmit Characteristic:
  - Contention transmission through CSMA

S-MAC-Summary

- Advantage
  - Idle listening is reduced by sleep schedules
  - Time synchronization overhead may be prevented by sleep schedule announcements
- Disadvantage
  - Adaptive listening incurs overhearing or idle listening
  - Sleep and listen periods are predefined and constant

Collision Avoidance

- S-MAC is based on contention
- 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)
  - But we only use in-channel signaling
- Who should sleep?
  - All immediate neighbors of sender and receiver
- How long to sleep?
  - The *duration* field in each packet informs other nodes the sleep interval

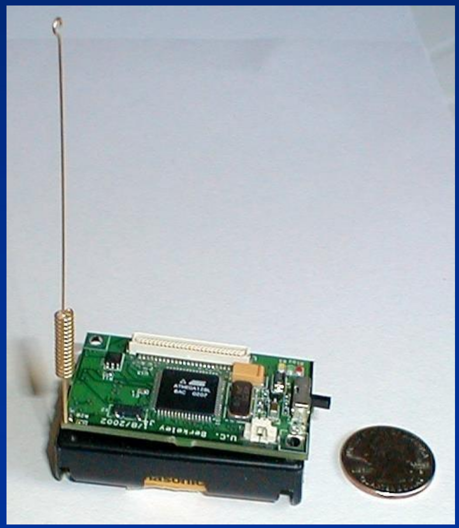
Message Passing

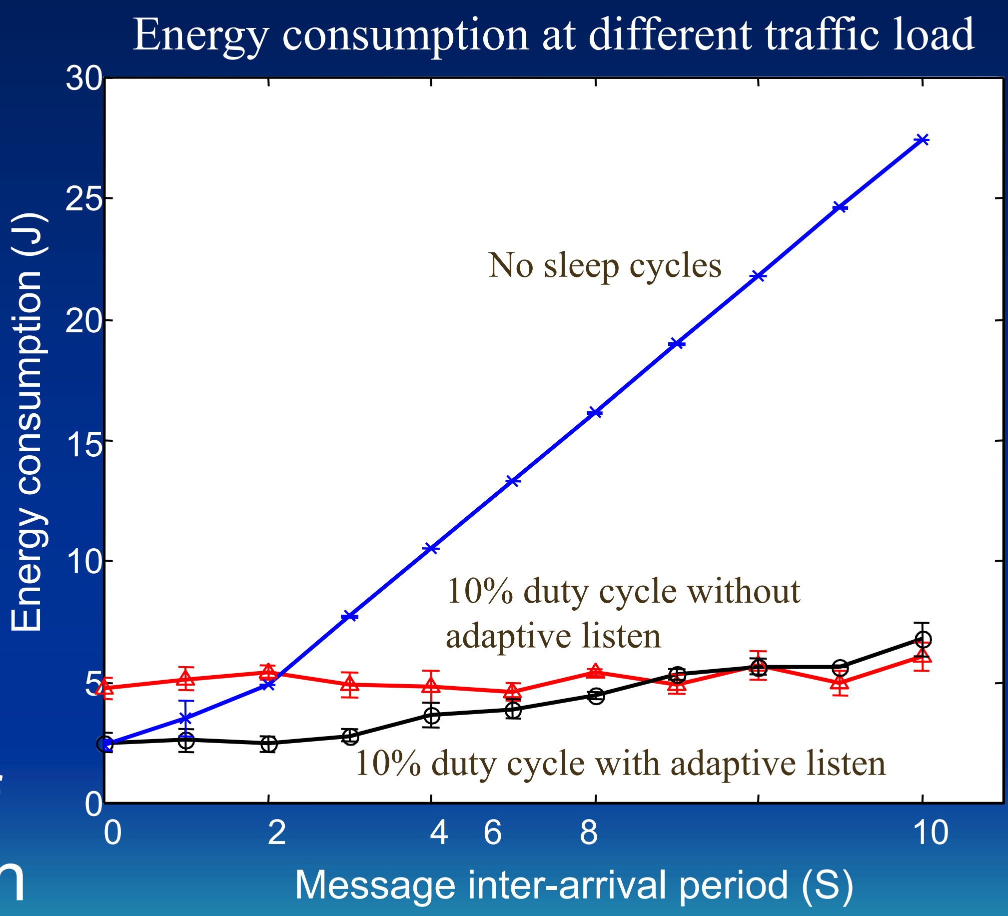
- Problem: Sensor net in-network processing requires *entire* message
- Solution: Don’t interleave different messages
  - Long message is fragmented & sent in burst
  - RTS/CTS reserve medium for entire message
  - Fragment-level error recovery — ACK
    — extend Tx time and re-transmit immediately
- Other nodes sleep for whole message time

Fairness → Energy
Msg-level latency

Implementation and Experiments

- Platform: Mica Motes
- Topology: 10-hop linear network





- S-MAC saved a lot of energy compared with a MAC without sleep

Case Study 2: B-MAC

- Another low-power MAC for sensor networks
- B-MAC design considerations
  - Simplicity: based on simple CSMA
  - Configurable options
  - Minimize idle listening
  - Based on model of periodic sensor data transfer
- B-MAC components
  - CSMA without RTS/CTS
  - Optional Low-power listening (LPL)
  - Optional ACK

Comparison of S-MAC and B-MAC

	S-MAC	B-MAC
Collision avoidance	CSMA/CA	CSMA
ACK	Yes	Optional
Message passing	Yes	No
Overhearing avoidance	Yes	No
Listen period	Pre-defined + adaptive listen	Pre-defined
Listen interval	Long	Very short
Schedule synchronization	Required	Not required
Packet transmission	Short preamble	Long preamble
Code size	6.3KB	4.4KB (LPL & ACK)

Next…

- Introduction to MAC
- MAC attributes and trade-offs
- Scheduled MAC protocols
- Contention-based MAC protocols
- Case studies
- Summary

MAC Design for Sensor Networks

- MAC protocols can be classified as scheduled and contention-based
- Major considerations
  - Energy efficiency
  - Scalability and adaptivity to number of nodes
- Major ways to conserve energy
  - Low duty cycle to reduce idle listening
  - Effective collision avoidance
  - Overhearing avoidance
  - Reducing control overhead

Scheduled vs. Contention Protocols

	Scheduled Protocols	Contention Protocols
Collisions	No	Yes
Energy efficiency	Good	Need improvement
Scalability and adaptivity	Bad	Good
Multi-hop communication	Difficult	Easy
Time synchronization	Strict	Loose or not required