Outline

- Background in MAC protocols
  - Role and features of MAC protocols
  - Scheduled access vs. Randomized access
  - Additional WSN requirements
- Two Examples
  - S-MAC
  - BMAC
Characteristics of Sensor Network

- 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
MAC and its Classification

- Medium Access Control (MAC)
  - When and how nodes access the shared channel

- Classification of MAC protocols
  - Scheduled protocols
    - Schedule nodes onto different sub-channels
    - Examples: TDMA, FDMA, CDMA
  - Contention-based protocols
    - Nodes compete in probabilistic coordination
    - Examples: ALOHA (pure & slotted), CSMA
MAC Attributes

- Collision avoidance
  - Basic task of a MAC protocol
- Energy efficiency
- Scalability and adaptivity
  - Network size, node density and topology change
- Channel utilization
- Latency
- Throughput
- Fairness
Energy Efficiency in MAC Design

- Energy is primary concern in sensor networks
- What causes energy waste?
  - Collisions
  - Control packet overhead
  - Overhearing unnecessary traffic
  - Long idle time
    - Bursty traffic in sensornet applications
    - Idle listening consumes 50—100% of the power for receiving
  - Wakeup period = sleep period + listen period
  - Duty Cycle = listen period/Wakeup period
Scheduled Protocols

- **TDMA**

  - **Advantages**
    - No collisions
    - Energy efficient — easily support low duty cycles

  - **Disadvantages**
    - Bad scalability and adaptivity
      - Difficult to accommodate node changes
      - Difficult to handle inter-cluster communication
    - Requires time synchronization
Scheduled Protocols

- Polling
  - A master plus one or more slaves (star topology)
  - The master node decides which slave can send by polling the corresponding slave
  - Only direct communication between the master and a slave
  - A special TDMA without pre-assigned slots
- Examples
  - IEEE 802.11 infrastructure mode (CFP)
  - Bluetooth piconets
Contention-Based Protocols

- Contention-based protocols
  - CSMA — Carrier Sense Multiple Access
    - Listening before transmitting
    - Collisions can still occur
  - CSMA/CA (CA stands for Collision Avoidance)
    - RTS/CTS handshake before send data
    - Other nodes (e.g., node c) delay transmission

Hidden terminal: a is hidden from c’s carrier sense
Examples of Contention-Based MACs

- **MACA** — Multiple Access w/ Collision Avoidance
  - Add duration field in RTS/CTS informing other nodes about their delay time

- **MACAW** — improved over MACA
  - RTS/CTS/DATA/ACK
  - Fast error recovery at link layer
# Scheduled vs. Contention Protocols

<table>
<thead>
<tr>
<th></th>
<th>Scheduled Protocols</th>
<th>Contention Protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collisions</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Energy efficiency</td>
<td>Good</td>
<td>Bad</td>
</tr>
<tr>
<td>Scalability and adaptivity</td>
<td>Bad</td>
<td>Good</td>
</tr>
<tr>
<td>Multi-hop communication</td>
<td>Difficult</td>
<td>Easy</td>
</tr>
<tr>
<td>Time synchronization</td>
<td>Strict</td>
<td>Loose or not required</td>
</tr>
</tbody>
</table>
Case Study: S-MAC

- S-MAC — by Ye, Heidemann and Estrin

- Tradeoffs
  - Increase latency and decrease fairness to improve energy efficiency

- Major components in S-MAC
  - Periodic listen and sleep
  - Collision avoidance
  - Overhearing avoidance
  - Massage passing
Coordinated Sleeping

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

  - Turn off radio when sleeping
  - Reduce duty cycle to ~ 10% (120ms on/1.2s off)

![Diagram showing listen and sleep cycles with latency and energy trade-off]
Coordinated Sleeping

- Schedules can differ

Choose neighboring nodes to have the same schedule—easy broadcast and low control overhead.

Border nodes: two schedules
Coordinated Sleeping

- **Schedule Synchronization**
  - New node tries to follow an existing schedule
  - 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

- **Periodic neighbor discovery**
  - Keep awake in a full sync interval over long periods
Coordinated Sleeping

- **Adaptive listening**
  - Reduce multi-hop latency due to periodic sleep
  - Wake up for a short period of time at end of each transmission

- Reduces latency by at least half
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
Implementation on Testbed Nodes

- Platform
  - Mica Motes (UC Berkeley)
    - 20Kbps radio at 433MHz

- Configurable S-MAC options
  - Low duty cycle with adaptive listen
  - Low duty cycle without adaptive listen
  - Fully active mode (no periodic sleeping)
Experiments: two-hop network

- **Topology and measured energy consumption on source nodes**
  - Source 1 -> Sink 1
  - Source 2 -> Sink 2

- S-MAC consumes much less energy than 802.11-like protocol w/o sleeping
- At heavy load, overhearing avoidance is the major factor in energy savings
- At light load, periodic sleeping plays the key role

![Average energy consumption in the source nodes](image)

Energy consumption (mJ) vs. Message inter-arrival period (second)

- 802.11-like protocol without sleep
- No periodic sleep
- S-MAC periodic sleep
Energy Consumption over Multi-Hops

- Ten-hop linear network at different traffic load

- At light traffic load, periodic sleeping has significant energy savings over fully active mode

- Adaptive listen saves more at heavy load by reducing latency
Latency as Hops Increase

- Adaptive listen significantly reduces latency caused by periodic sleeping.

![Graph showing latency under lowest traffic load with and without adaptive listen, and with and without sleep cycles.]

![Graph showing latency under highest traffic load with and without adaptive listen, and with and without sleep cycles.]
Throughput as Hops Increase

- Adaptive listen significantly increases throughput
  - Using less time to pass the same amount of data
Combined Energy and Throughput

- Periodic sleeping provides excellent performance at light traffic load
- With adaptive listening, S-MAC achieves about the same performance as no-sleep mode at heavy load
Adaptive Listen Slots

- In S-MAC all nodes have listen slots of the same duration
  - Different nodes might have different tx/rx patterns
  - Idle listening wastes power
  - Idea: adaptively change the idle listen slot

Low Power Listening

- **Goal:** minimize listen cost

- **Principles**
  - Node periodically wakes up, turns radio on and checks channel
    - **Wakeup time fixed**
    - **“Check time” variable**
  - If energy is detected, node powers up in order to receive the packet
  - Node goes back to sleep
    - If a packet is received
    - After a timeout
  - Preamble length matches channel checking period
    - **No explicit synchronization required**
  - **Noise floor estimation used to detect channel activity during LPL**
Radio power up sequence of operations

- **Goals**
  - Minimize time radio is on
  - Minimize number of times radio gets started
  - Minimize sampling time (stage e)
LPL check interval

- Single-hop application doing periodic data sampling
- Sampling rate (traffic pattern) defines optimal check interval
- Check interval
  - Too small: energy wasted on idle listening
  - Too large: energy wasted on transmissions (long preambles)
- In general, it’s better to have larger preambles than to check more often!