MAC Layer Design for Wireless Sensor Networks

Wei Ye
USC Information Sciences Institute
Updated by Ajay Gupta, WMU

Characteristics of Sensor Network

• A special wireless ad hoc network
  – Large number of nodes ➔ Scalability & Self-configuration
  – Battery powered ➔ Energy efficiency
  – Topology and density change ➔ Adaptivity
  – Nodes for a common task ➔ Fairness not important
  – In-network data processing ➔ Message-level Latency
• Sensor-net applications
  – Sensor-triggered bursty traffic ➔ Adaptivity
  – Can often tolerate some delay ➔ Trade for energy
  • 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 sensor-net apps
    - Idle listening consumes 50—100% of the power for receiving (Stemm97, Kasten)

Contestation-Based Protocols

- Contention-based protocols
  - CSMA — Carrier Sense Multiple Access
    - Listening before transmitting
    - Not enough for multi-hop networks (collision at receiver)
  - CSMA/CA (CA stands for Collision Avoidance)
    - RTS/CTS handshake before send data
    - Other nodes (e.g. node c) backoff

Hidden terminal: a is hidden from c’s carrier sense
Hidden Terminal Problem

- A and C cannot hear each other.
- A sends to B, C cannot receive A.
- C wants to send to B, C senses a "free" medium (CS fails).
- Collision occurs at B.
- A cannot receive the collision (CD fails).
- A is "hidden" for C.

Exposed Terminal Problem

- A starts sending to B.
- C senses carrier, finds medium in use and has to wait for A->B to end.
- D is outside the range of A, therefore waiting is not necessary.

Solution for Hidden Terminals

- A first sends a Request-to-Send (RTS) to B
- On receiving RTS, B responds Clear-to-Send (CTS)
- Hidden node C overhears CTS and keeps quiet
  - Transfer duration is included in both RTS and CTS
- Exposed node overhears a RTS but not the CTS
  - D's transmission cannot interfere at B
Contention-Based Protocols

- Contention-based protocols (contd.)
  - MACA — Multiple Access w/ Collision Avoidance
    - Add duration field in RTS/CTS informing other node about their backoff time
  - MACAW — improved over MACA
    - RTS/CTS/DATA/ACK
    - Fast error recovery at link layer
  - IEEE 802.11 Distributed Coordination Function (DCF)
    - Largely based on MACAW

Contention-Based Protocols

- IEEE 802.11 DCF: ad hoc mode
  - 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-Based Protocols

- Tx rate control — by Woo and Culler
  - Based on a special network setup
    - A base station tries to collect data equally from all sensors in the network
  - CSMA + adaptive rate control
  - Promote fair bandwidth allocation to all sensors
    - Nodes close to the base station forward more traffic, and have less chances to send their own data
  - Helps in congestion avoidance
### 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>

### Energy Efficiency in Contention Protocols
- Contention-based protocols need to work hard in all directions for energy savings
  - Reduce idle listening – support low duty cycle
  - Better collision avoidance
  - Reduce control overhead
  - Avoid unnecessary overhearing

### Energy-Efficient MAC Design
- PAMAS: Power Aware Multi-Access with Signaling — by Singh and Raghavendra
  - Improve energy efficiency from MACA
  - Avoid overhearing by putting node into sleep
  - Use separate control and data channels
    - RTS, CTS, busy tone to avoid collision
    - Probe packets to find neighbors transmission time
  - Increased hardware complexity
    - Two channels need to work simultaneously, meaning two radio systems.
Energy-Efficient MAC Design

- Piconet — by Bennett, Clarke, et al.
  - Not the same piconet in Bluetooth
  - Low duty-cycle operation — energy efficient
    - Sleep for 30s, beacon, and listen for a while
    - Sending node needs to listen for receiver’s beacon first, then
    - CSMA before sending data
  - May wait for long time before sending

Energy-Efficient MAC Design

- 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

Energy-Efficient MAC Design

- Unicast example of PS mode in 802.11 DCF
Energy-Efficient MAC Design

- Asynchronous sleeping – by Tseng, et al.
  - Extend 802.11 PS mode to Multi-hops
  - Nodes do not synchronize with each other
  - Designed 3 sleep patterns — ensure nodes listen intervals overlap, example:
    - Periodically fully-awake interval: similar to S-MAC
  - Problem on broadcast — wake up each neighbor

Energy-Efficient MAC Design

- ZigBee
  - Industry standard through application profiles running over IEEE 802.15.4 radios
  - Target applications are sensors networks, interactive toys, smart badges, remote controls, and home automation

Energy-Efficient MAC Design

- ZigBee (Cont.)
  - Three devices specified
    - 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
Case Study: S-MAC

- S-MAC — by Ye, Heidemann and Estrin
- Tradeoffs
  - Latency
  - Fairness
  - Energy

- 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)

Coordinated Sleeping

- Schedules can differ
  - Prefer neighboring nodes have same schedule — easy broadcast & low control overhead
  - Border nodes: two schedules or broadcast twice
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

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**

Implementation on Testbed Nodes

- **Platform**
  - Mica Motes (UC Berkeley)
    - 8-bit CPU at 4MHz
    - 128KB flash, 4KB RAM
    - 20Kbps radio at 433MHz
  - TinyOS: event-driven
- **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
  - Source 2
  - Sink 1
  - 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

Energy Consumption over Multi-Hops

- Ten-hop linear network at different traffic load
- 3 configurations of S-MAC
- 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
**Throughput as Hops Increase**

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

```
Effective data throughput under highest traffic load
```

- Number of hops
- Effective data throughput (Byte/S)
- No sleep cycles
- 10% duty cycle with adaptive listen
- 10% duty cycle without adaptive listen

**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

```
Energy-time product per byte (J*S/byte)
```

- Message inter-arrival period (S)
- No sleep cycles
- 10% duty cycle without adaptive listen
- 10% duty cycle with adaptive listen