S-MAC
Sensor Medium Access Control Protocol

An Energy Efficient MAC protocol for Wireless Sensor Networks

Outline

- Introduction
- Design Considerations
- Main sources of energy inefficiency
- Current MAC design
- S-MAC
- Protocol implementation in a test-bed
- Discussion
- Conclusion and future work

Wireless Sensor Networks

- Application specific wireless networks for monitoring, smart spaces, medical systems and robotic exploration
- Battery operated and power limited sensor devices
- Large number of distributed nodes deployed in an ad-hoc fashion

Design Considerations

Primary attributes:

- Energy Efficiency

  often difficult to recharge or replace batteries

  prolonging the network lifetime is important

- Scalability

  Some nodes may die or new nodes may join

Secondary attributes:

Fairness, latency, throughput and bandwidth

Sources of Energy Inefficiency

- Collision
- Overhearing
- Control packet overhead
- Idle listening

Existing MAC Design

- Contention-based protocols
  - IEEE 802.11 – Idle listening
  - PAMAS – heavy duty cycle of the radio, avoids overhearing, idle listening
- TDMA based protocols

  Advantages - Reduced energy consumption

  Problems – requires real clusters,

  and does not support scalability

Design goal of S-MAC

- Reduce energy consumption
- Support good scalability
- Self-configurable

S-MAC

- Tries to reduce wastage of energy from all four sources of energy inefficiency
  - Collision – by using RTS and CTS
  - Overhearing – by switching the radio off when transmission is not meant for that node
  - Control Overhead – by message passing
  - Idle listening – by periodic listen and sleep

Is the improvement free of cost?

- No
- In exchange there is some reduction in per-hop fairness and latency
- But does not reduce end-to-end fairness and latency

Is it important for sensor networks?

Network Assumptions

- Composed of many small nodes deployed in ad hoc fashion
- Most communication will be between nodes as peers, rather than a single base station
- Nodes must self-configure

Application Assumptions

- Dedicated to a single application or a few collaborative application
- Involves in-network processing to reduce traffic and increase life time
- Applications will have long idle periods and can tolerate some latency

Components of S-MAC

- Periodic listen and sleep
- Collision and Overhearing avoidance
- Message passing

Periodic Listen and Sleep

- Each node goes into periodic sleep mode during which it switches the radio off and sets a timer to awake later
- When the timer expires, it wakes up
- Selection of sleep and listen duration is based on the application scenarios
- Neighboring nodes are synchronized together

Contd….

- Nodes exchange schedules by broadcast
- Multiple neighbors contend for the medium
- Once transmission starts, it does not stop until completed

Choosing and Maintaining Schedules

- Each node maintains a schedule table
- Initial schedule is established
  - Synchronizer
  - Follower

- Rules for joining a new node

Maintaining Synchronization

- Needed to prevent clock drift
- Periodic updating using a SYNC packet

Sender Node ID	Next-Sleep Time

SYNC Packet

- Receivers adjust their timer counters
- Listen interval divided into two parts

Each part further divided into time slots

Timing Relationship

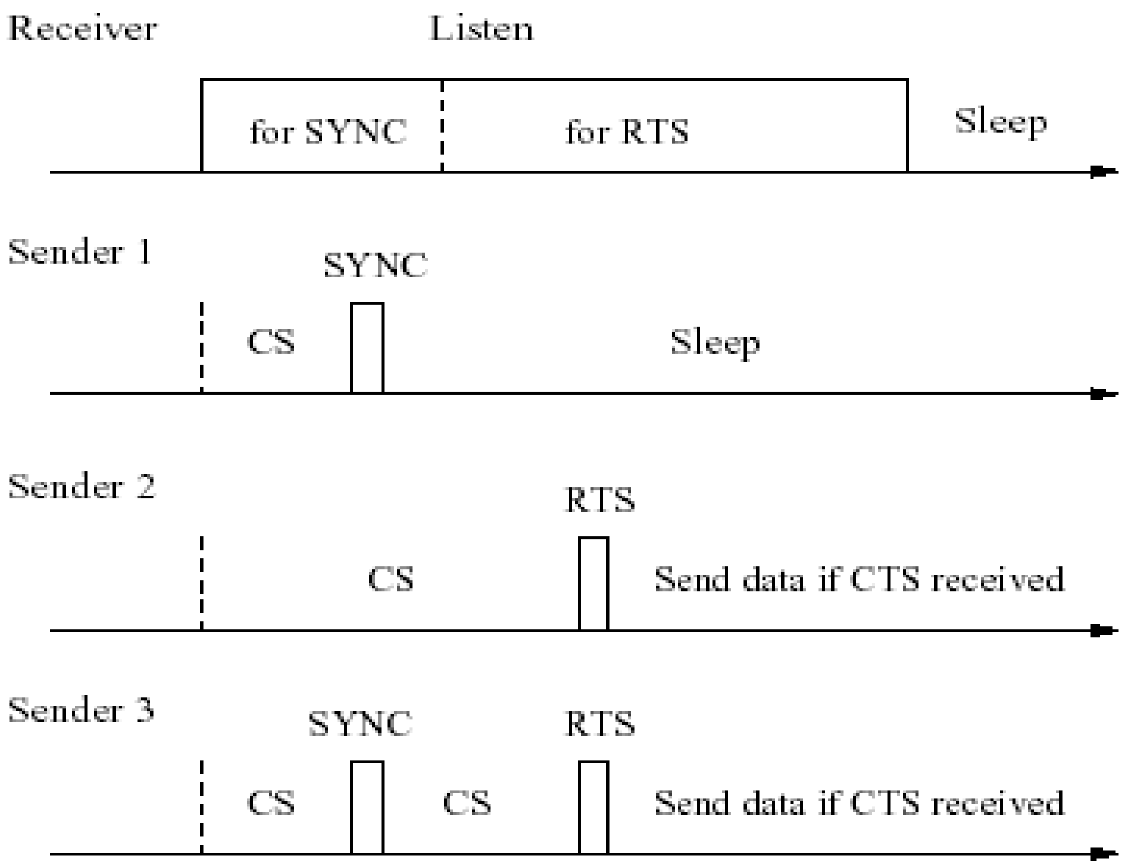

Fig. 3. Timing relationship between a receiver and different senders. CS stands for carrier sense.

Collision Avoidance

- Similar to IEEE 802.11 using RTS/CTS mechanism
- Perform virtual and physical carrier sense before transmission
- RTS/CTS addresses the hidden terminal problem
- NAV –indicates how long the remaining transmission will be.

Overhearing Avoidance

- Interfering nodes go to sleep after they hear the RTS or CTS packet
- The medium is busy when the NAV value is not zero
- All immediate neighbors of sender and receiver should go to sleep

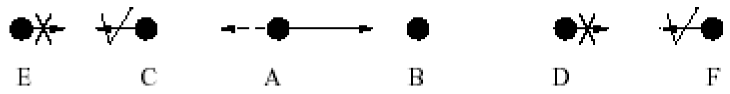

Fig. 4. Who should sleep when node A is transmitting to B?

Message Passing

- What is a message?
- Transmitting a message as a long packet

  High retransmission cost
- Fragmentation into small packets

  High control overhead

➢ Solution

➢ Disadvantage

Protocol Implementation

- Test bed
  - Rene motes developed at UCB
  - They run TinyOS, an event–driven operating system
- Two types of packets
  - Fixed size data packets with header(6B), payload(30B) and CRC(2B)
  - Control packets (RTS and CTS), header(6B) (2B) CRC

MAC modules implemented

- Simplified IEEE 802.11 DCF – physical and virtual carrier sense, backoff and retry, RTS/CTS/DATA/ACK packet exchange and fragmentation support
- Message passing with overhearing avoidance
- The complete S-MAC – all the features are implemented

Conclusions and Future work

- S-MAC has good energy conserving properties comparing to IEEE 802.11

Future work

- Analytical study on the energy consumption and latency
- Analyze the effect of topology changes

Our Project

- Implementing S-MAC on TinyOS 1.0
- Incorporating multicasting with S-MAC

Directed Diffusion and S-MAC

- S-MAC can be incorporated into the directed diffusion paradigm