An Energy-efficient MAC protocol for Wireless Sensor Networks

Wei Ye, John Heidemann, Deborah Estrin

presented by

Venkatesh Rajendran

Outline

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

Wireless Sensor Networks

- Application specific wireless networks for monitoring, smart spaces, medical systems and robotic exploration
- Large number of distributed nodes and self organizing
- Normally battery operated and hence power limited

Design Considerations

- Energy efficiency
  - often difficult recharge batteries or replace them
  - prolonging the life-time is important
- Scalability to the change in network size, node density and topology
  - some nodes may die over time
  - new nodes may join later

Design Considerations

- Other important attributes
  - Fairness
  - Latency
  - Throughput
  - Bandwidth Utilization
- These are generally the primary concerns in traditional wireless voice and data networks
- But in sensor networks they are secondary

Sources of Energy Inefficiency

- Collision
  - corrupted packets must be retransmitted and it increases energy consumption.

- Overhearing
  - picking up packets that are destined to other nodes

Sources of Energy Inefficiency

- Control packet overhead
- Idle listening
  - Listening to receive possible traffic that is not sent
  - This is the major source of energy inefficiency
  - consumes 50-100% of the energy required for receiving

Current MAC Design

- Contention based protocols
  - IEEE 802.11 distributed coordination function (DCF) - high energy consumption due to idle listening
  - PAMAS
    - avoids the overhearings among neighboring nodes
    - requires two independent radio channels
    - does not address the issue of reduce idle listening

TDMA based protocols

- Advantages
  - lower energy conservation when compared to contention based as the duty cycle of the radio is reduced and no contention overhead
- Problems
  - Requires nodes to form real communication clusters and managing inter-cluster communication is difficult
  - It is not easy to change the slot assignment dynamically, hence scalability is not as good as contention based

Design goal of S-MAC

- Reduce energy consumption
- Support good scalability and collision avoidance

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 the 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 both per-hop fairness and latency
- This does not necessarily result in lower end-to-end fairness and latency

Per-hop fairness

- It is important in wireless voice or data networks as each user desires equal opportunity and time to access the network
- Is it important for sensor networks?
  - In sensor networks all nodes co-operate and work together for a single application
  - So per-hop fairness is not important as long as application level performance is not degraded.

Network assumptions

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

Application assumptions

- Dedicated to a single application or a few collaborative applications
- Involves in-network processing to reduce traffic and thereby increase the life-time
- This implies that data will be processed as whole messages at a time in store-and-forward fashion
- Hence packet or fragment-level interleaving from multiple sources only delays overall latency
- Applications will have long idle periods and can tolerate some latency

Features of S-MAC

The main features of S-MAC are:

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

Periodic Listen and Sleep

- If no sensing event happens, nodes are idle for a long time
- So it is not necessary to keep the nodes listening all the time
- Each node go 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 and listens to see if any other node wants to talk to it

Fig. 1. Periodic listen and sleep.

- Duration of sleep and listen time can be selected based on the application scenario
- To reduce control overhead, neighboring nodes are synchronized (i.e. Listen and sleep together)

Fig. 2. Neighboring nodes A and B have different schedules. They synchronize with nodes C and D respectively.

- Not all neighboring nodes can synchronize together
- Two neighboring nodes (A and B) can have different schedules if they are required to synchronize with different node

- If a node A wants to talk to node B, it just waits until B is listening
- If multiple neighbors want to talk to a node, they need to contend for the medium
- Contention mechanism is the same as that in IEEE802.11 (using RTS and CTS)
- After they start data transmission, they do not go to periodic sleep until they finish transmission

Choosing and Maintaining Schedules

- Each node maintains a schedule table that stores schedules of all its known neighbors.
- To establish the initial schedule (at the startup) following steps are followed:
  - A node first listens for a certain amount of time.
  - If it does not hear a schedule from another node, it randomly chooses a schedule and broadcast its schedule immediately.
  - This node is called a SYNCHRONIZER.

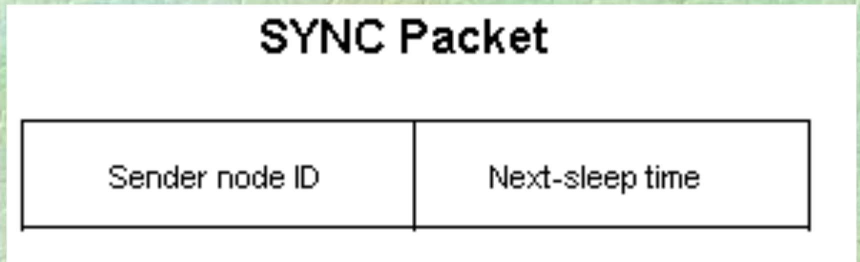
- If a node receives a schedule from a neighbor before choosing its own schedule, it just follows this neighbor's schedule.
- This node is called a FOLLOWER and it waits for a random delay and broadcasts its schedule.
- If a node receives a neighbor's schedule after it selects its own schedule, it adopts to both schedules and broadcasts its own schedule before going to sleep.

Rules for Joining a New Node

- Listen for a long time until an active node is discovered
- Send INTRO packet to the active node
- Active node forwards its schedule table
- Treat all the nodes on table as potential neighbors and contact them later
- If possible follow the synchronizer's schedule else establish a random schedule and broadcast the schedule

Maintaining Synchronization

- Timer synchronization among neighbors are needed to prevent the clock drift.
- Done by periodic updating using a SYNC packet.
- Updating period can be quite long as we don't require tight synchronization.
- Synchronizer needs to periodically send SYNC to its followers.
- If a follower has a neighbor that has a different schedule with it, it also needs update that neighbor.



- Time of next sleep is relative to the moment that the sender finishes transmitting the SYNC packet
- Receivers will adjust their timer counters immediately after they receive the SYNC packet
- Listen interval is divided into two parts: one for receiving SYNC and other for receiving RTS

Timing Relationship of Possible Situations

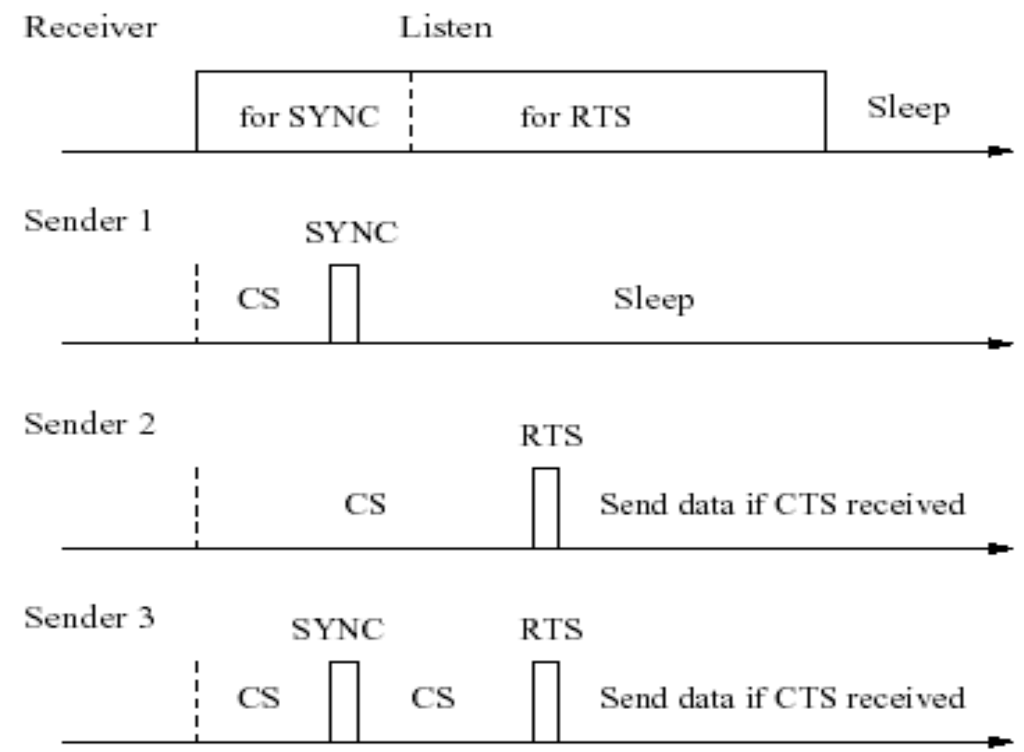


Fig. 3. Timing relationship between a receiver and different senders. CS stands for carrier sense.

Collision Avoidance

- Similar to IEEE802.11 using RTS/CTS mechanism
- Perform carrier sense before initiating a transmission
- If a node fails to get the medium, it goes to sleep and wakes up when the receiver is free and listening again
- Broadcast packets are sent without RTS/CTS
- Unicast packets follow the sequence of RTS/CTS/ DATA/ACK between the sender and receiver

Overhearing Avoidance

- Duration field in each transmitted packet indicates how long the remaining transmission will be.
- So if a node receives a packet destined o another node, it knows how long it has to keep silent.
- The node records this value in network allocation vector (NAV) and set a timer.

- When a node has data to send, it first looks at NAV.
- If NAV is not zero, then medium is busy (virtual carrier sense).
- Medium is determined as free if both virtual and physical carrier sense indicate the medium is free.
- All immediate neighbors of both the sender and receiver should sleep after they hear RTS or CTS packet until the current transmission is over.

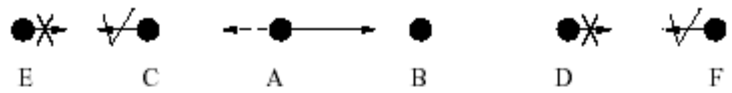


Fig. 4. Who should sleep when node A is transmitting to B?

Message Passing

- A message is a collection of meaningful, interrelated units of data
- Transmitting a long message as a packet is disadvantageous as the re-transmission cost is high
- Fragmentation into small packets will lead to high control overhead as each packet should contend using RTS/CTS

Solution

- Fragment message in to small packets and transmit them as a burst
- Advantages
  - Reduces latency of the message
  - Reduces control overhead
- Disadvantage
  - Node-to-node fairness is reduced, as nodes with small packets to send has to wait till the message burst is transmitted

Protocol Implementation

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

Topology

- Two-hop network with two sources and two sinks
- Sources generate message which is divided into fragments
- Traffic load is changed by varying the inter-arrival period of the message

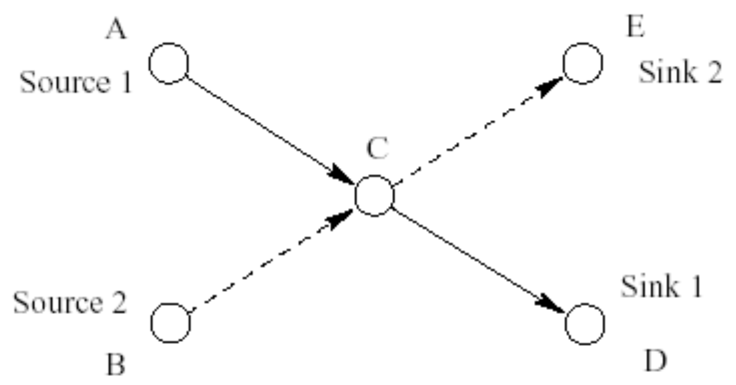


Fig. 6. Topology used in experiments: two-hop network with two sources and two sinks.

Energy consumption in the source nodes

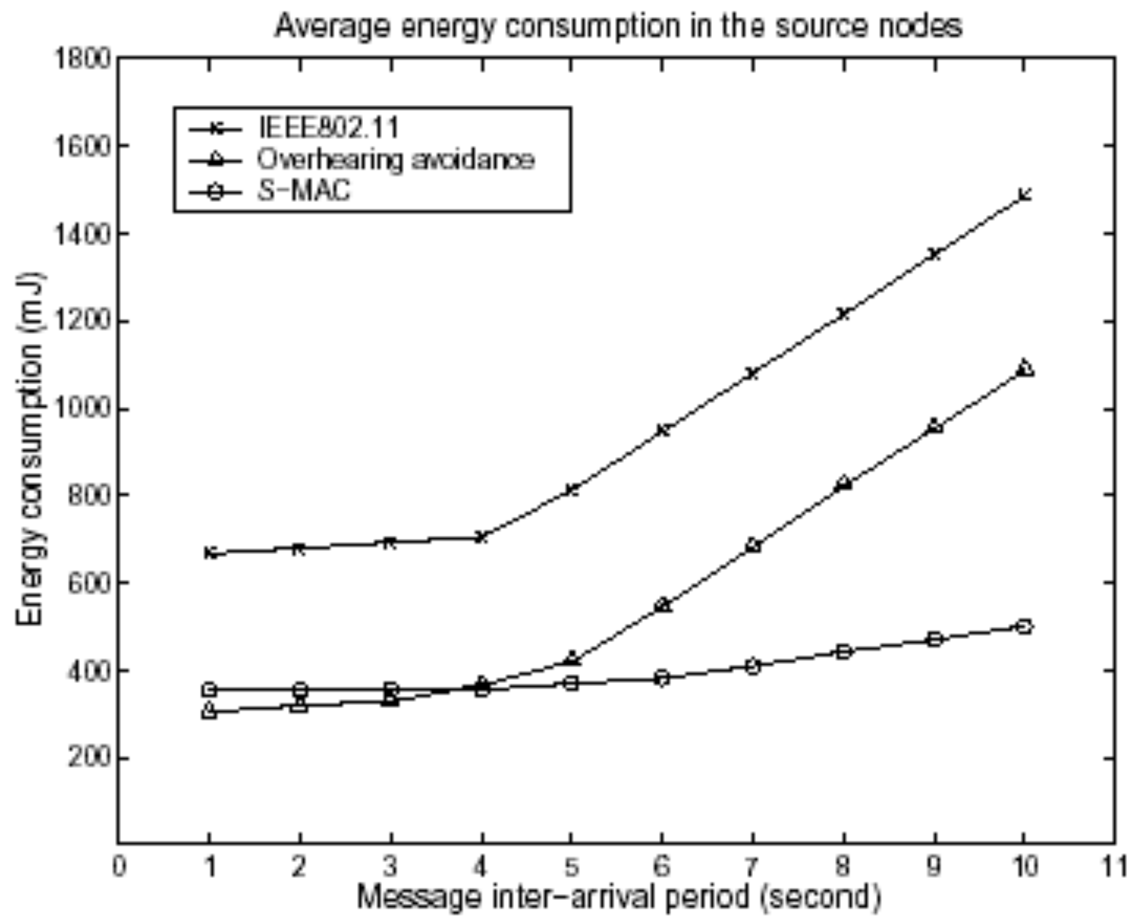


Fig. 7. Measured energy consumption in the source nodes.

Percentage of time that the source nodes are in the sleep mode

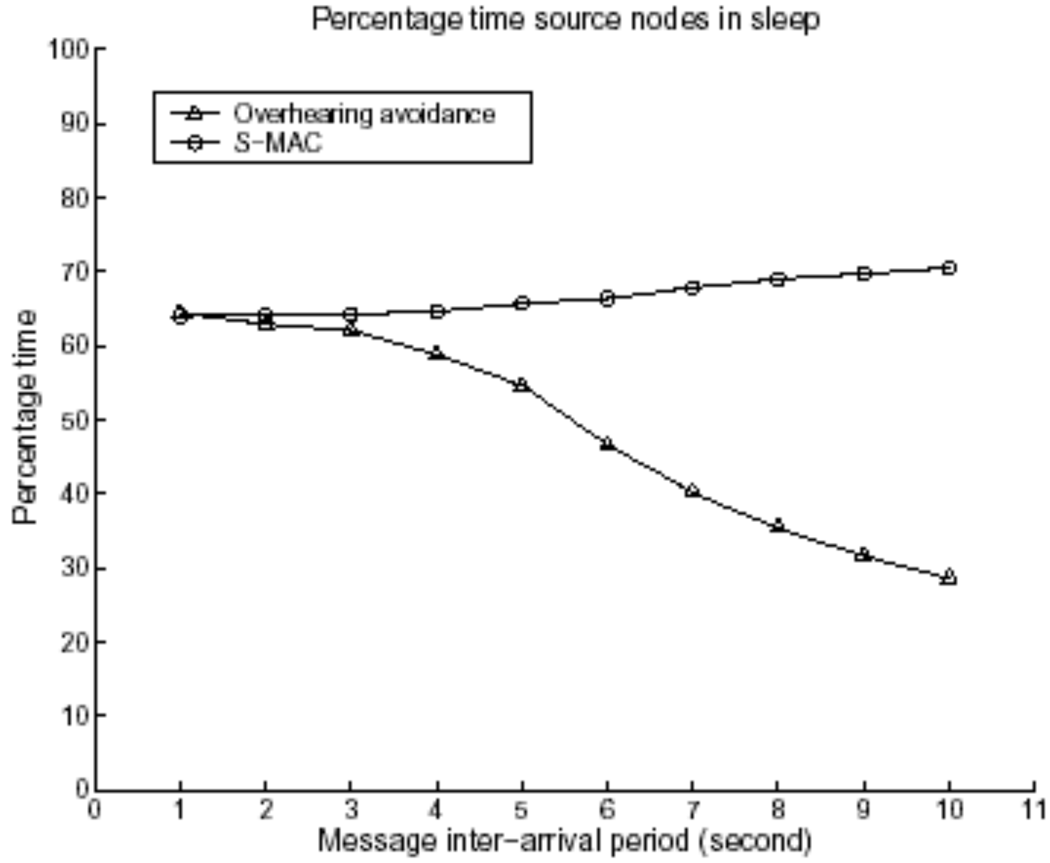


Fig. 8. Measured percentage of time that the source nodes in the sleep mode.

Energy consumption in the intermediate node

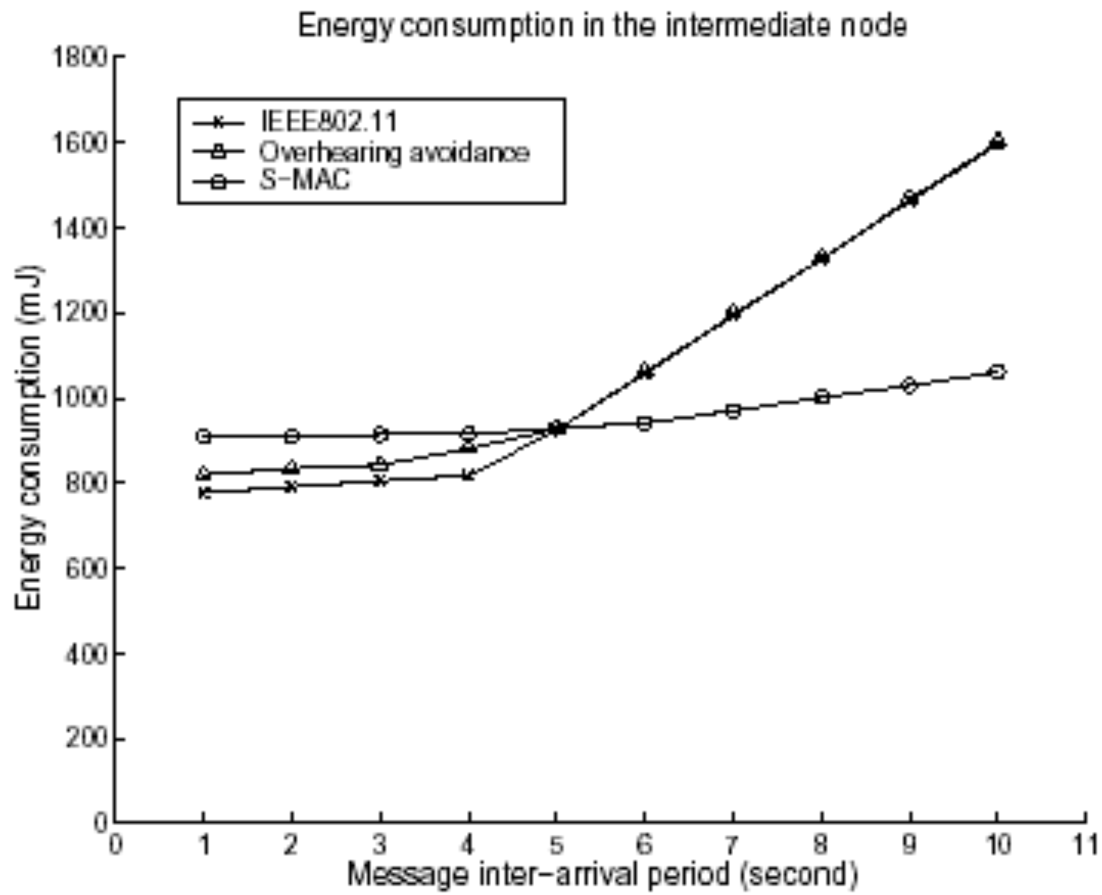


Fig. 9. Measured energy consumption in the intermediate node.

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