S-MAC

Xuan Zhong
School of Electrical and Computer Engineering
Purdue University

Outline

- Introduction
- Protocol Design
- Architecture Design and Implementation
- Experimentation Results
- Conclusions

Introduction

- S-MAC design goal
  - To reduce energy consumption
- Main sources of energy wastes
  - Collision
  - Overhearing
  - Control packet overhead
  - Idle listening

Message passing

Coordinated sleeping among neighboring nodes

Periodically putting nodes to sleep

Protocol Design

- Assumptions
  - Large number of small nodes
  - Ad-hoc topology
  - Data being processed as whole messages in a store-and-forward fashion
- Applications
  - Have long idle periods (until some event is detected)
  - Can tolerate some additional latency
  - Network lifetime is critical

Protocol Design

A. Nodes coordinate their sleep schedules rather than randomly sleep on their own

- Each node maintains a schedule table that stores all its known neighbors
- A node first listens for a fixed amount of time, which is at least the synchronization period. If it does not hear a schedule from another node, it immediately choose its own schedule and starts to follow it. Meanwhile, the node tries to announce the schedule by broadcasting a SYNC packet.
- If the node receives a schedule from a neighbor before choosing or announcing its own schedule, it follows that schedule by setting its schedule to be the same. Then the node will try to announce its schedule at its next scheduled listen time
- What if a node receives a different schedule after it chooses and announces its own schedule?
  - If the node has no other neighbors, it will discard its current schedule and follow the new one
  - If the node has already follows a schedule with one or more neighbors, it adopts both schedules by waking up at the listen intervals of the two schedules

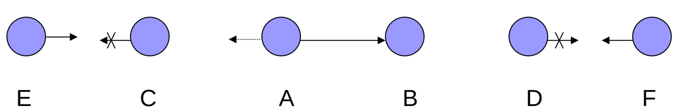
Protocol Design

- Maintain synchronization
  - Neighboring nodes maintain synchronization by exchanging relative timestamps and setting the listen period significantly longer than clock drift rates
- Adaptive listening – improve the latency caused by the periodic sleep
  - To let the 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 transmission.
  - If the node is the next-hop node, its neighbor is able to immediately pass the data to it instead of waiting for its scheduled listen time
  - If the node does not receive anything during the adaptive listening, it will go back to sleep until its next scheduled listen time

Protocol Design

B. Collision Avoidance

- Overhearing Avoidance



Rule: All immediate neighbors of both sender and receiver should sleep after hearing a RTS/CTS for the transmission duration

- Message Passing
  - Fragment the long message into many small fragments, and transmit them in a burst.
  - Only one RTS/CTS, and they reserve medium for all fragments.
  - Every fragment needs an ACK, if ACK not received, reserve medium for one more fragment and re-transmit current fragment.

S-MAC Implementation

PARAMETERS OF S-MAC IMPLEMENTATION ON MICA MOTES

Radio bandwidth	20Kbps
Channel coding	Manchester
Control packet length	10 bytes
Data packet length	up to 250 bytes
MAC header length	8 bytes
Duty cycle	1% to 99%
Duration of listen interval	115 ms
Contention window for SYNC	15 slots
Contention window for data	31 slots

S-MAC Stack

- **Layered structure** - Clearly separates the functions of the physical layer and MAC layer

The physical layer directly controls the radio and provides APIs for upper layers to put the radio into different states: sleep, idle, transmission and reception. It does start symbol detection, channel coding and decoding, byte buffering, and CRC check. It also provides the carrier sense functionality, but gives the full control to the MAC layer.

- **Nested header structure**

It allows each layer to freely define its own packet types as well as add its header fields to a packet coming from its upper layers.

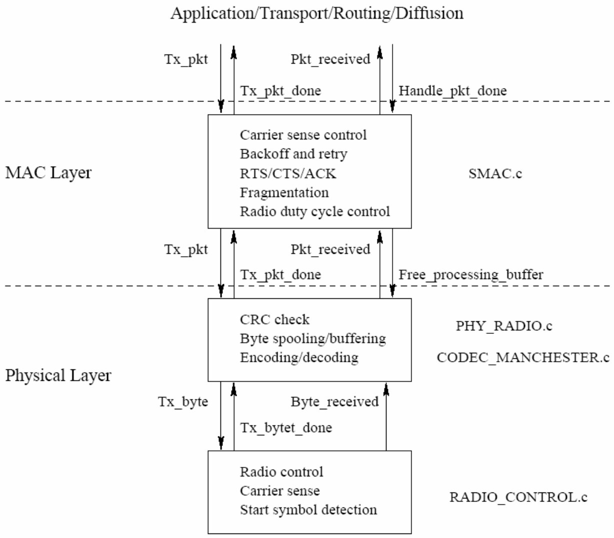


Figure 2: Stack structure and functions.

Picture source: Wei Ye, John Heidemann and Deborah Estrin, A Flexible and Reliable Radio Communication Stack on Motes. Technical Report ISI-TR-565, USC Information Sciences Institute, September 2002.

Experimental Results

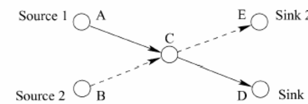


Fig. 7. Topology 1: two-hop network with two sources and two sinks.

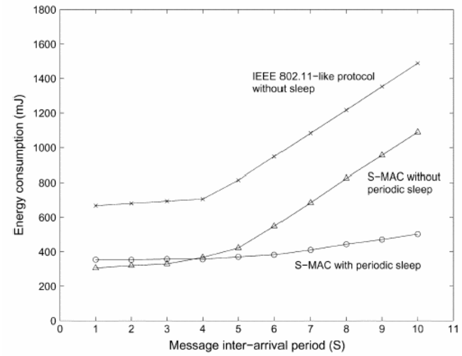


Fig. 8. Mean energy consumption on radios in each source node.

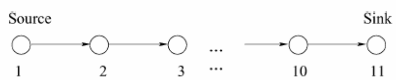


Fig. 9. Topology 2: ten-hop linear network with one source and one sink.

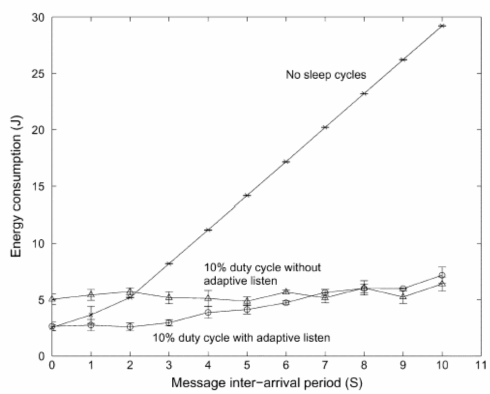


Fig. 10. Aggregate energy consumption on radios in the entire ten-hop network using three S-MAC modes.

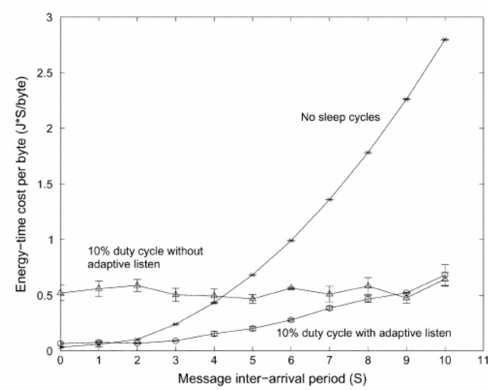


Fig. 15. Energy–time cost per byte on passing data from source to sink under different traffic load.

Conclusions

- S-MAC Attributes:
  - Nodes periodically sleep to reduce energy consumption in listening to an idle channel
  - Neighboring nodes form virtual clusters to auto-synchronize on sleep schedules
  - S-MAC uses message passing to reduce contention latency for applications that require store-and-forward processing
- Ability to tradeoff energy vs. latency
- 2-6 times energy savings compared to 802.11 for traffic load with messages sent every 1-10 seconds

Reference

(1). Wei Ye, John Heidemann and Deborah Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks, *IEEE/ACM Trans. on Networking*, vol. 12, no. 3, pp 493-506, June 2004.