Demonstration of Schedule and Latency Control in S-MAC

USC/ISI Technical Report ISI-TR-581, November 2003 *

Yuan Li, Wei Ye and John Heidemann

Information Science Institute
University of Southern California

S-MAC is a medium-access control (MAC) protocol designed for wireless sensor networks. It is different from traditional wireless MACs such as IEEE 802.11 in several ways: energy conservation and self-configuration are primary goals, while per-node fairness and latency are less important. S-MAC uses a few novel techniques to reduce energy consumption and support self-configuration. It enables low-duty-cycle operation in a multi-hop network. Nodes form virtual clusters based on common sleep schedules to reduce control overhead and enable traffic-adaptive wake-up. S-MAC uses in-channel signaling to avoid overhearing unnecessary traffic. Finally, S-MAC applies message passing to reduce contention latency for applications that require in-network data processing. Although S-MAC has been designed for sensor-network applications, its principles are applicable to any application where battery lifetime dominates other design goals. S-MAC has been previously described in two papers [1, 2].

S-MAC's peer-to-peer sleep/wakeup schedule is one of its most novel elements. While TDMA- and cluster-based sleep schedules are well understood, to our knowledge S-MAC is the first MAC protocol where individual nodes can select their own sleep schedules independently. This demonstration will examine schedule and latency control in S-MAC with Berkeley Motes [3].

The first part of the demonstration shows how S-MAC manages schedules in a multi-hop network. The first and simplest case is in a single-hop network, where all nodes can hear each other. We will demonstrate that S-MAC attempts to synchronize all nodes on the same schedule. The second case is in a large, multi-hop network, where nodes form different clusters by following different schedules. We will present that different nodes automatically configure their schedules and the network as a whole can support multiple schedules. In the last case, we will apply a new algorithm to allow nodes in multiple clusters to incrementally switch to one global schedule.

The second part of the demonstration shows the latency in data transmission in a multi-hop network. We will first demonstrate the transmission delay due to sleep schedules, and how S-MAC's adaptive listen technique reduces the delay by waking up the next-hop node in anticipation of message forwarding. Then we will demonstrate how specifically controlled sleep schedules affect latency. Data transfer in any network with sleep schedules can incur additional delay when a schedule slot is missed. A unique characteristic of a peer-to-peer network such as S-MAC is that we can select and control sleep schedules to obtain different effects. In fact, sleep schedules can be *skewed* to allow rapid data forwarding in one direction, and slow forwarding in the opposite direction. We will place nodes in a line and configure their schedules to allow rapid propagation in one direction and slow propagation in the opposite.

References

[1] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the IEEE Infocom*, pages 1567–1576, New York, NY, June 2002.

[2] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. Technical Report ISI-TR-567, USC Information Sciences Institute, January 2003. To appear in the IEEE/ACM Transactions on Networking.

[3] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for networked sensors. In *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 93–104, Cambridge, MA, USA, November 2000. ACM.

*This work was presented as a demo at ACM SenSys 2003, November 2003

Schedule and Latency Control in S-MAC

Yuan Li, Wei Ye, John Heidemann
University of Southern California, Information Sciences Institute
ISI Laboratory for Embedded Networked Sensor Experimentation, http://www.isi.edu/ilense/

Introduction: S-MAC

S-MAC

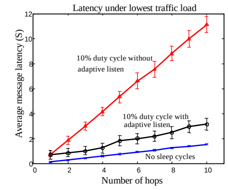
- Medium-access control (MAC) protocol for wireless sensor networks
- Primary goals: *energy conservation* and *self-configuration*
- Low-duty-cycle operation in a multi-hop network
- Nodes form virtual clusters on sleep schedules
- Uses in-channel signaling to avoid overhearing
- Uses Message passing to reduce contention latency

Schedules in S-MAC

- Nodes adopt listen/sleep cycle to conserve energy
- Nodes coordinate on their sleep schedules (rather than waking up randomly)
- Schedules should be synchronized to minimize latency

Latency in S-MAC

- Duty cycling can increase latency
- Can trade off latency and fairness for energy savings

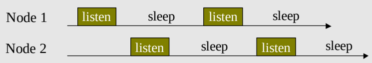


In all three S-MAC modes, latency increases linearly with the number of hops

Challenges: Schedule and Latency Control in S-MAC

Multiple Schedules on Border Nodes

- Nodes automatically configure schedules
- Nodes form virtual clusters, multiple schedules
- Border nodes wake up more frequently and consume more energy
- Can select single global schedule



Multiple schedules

Applications have Different Latency Requirements

- Different applications require different latencies on data delivery
- Urgent data need to be transferred quickly
- Can control schedules to get different latency effects

Approaches: Global Schedule and Latency Control by Adjusting Schedules

Selecting Global Schedule

Goal:

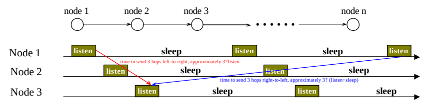
Nodes in multiple clusters can incrementally switch to one global schedule

Algorithm:

- Assign unique schedule id (randomly)
- Nodes incrementally shift schedules
  - Prefer schedule with lowest id
- Over time, all nodes shift to a single global schedule

Control Sleep Schedules

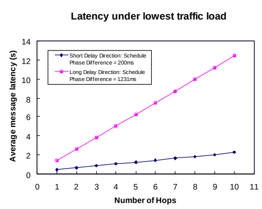
- Select and control sleep schedules to obtain different effects on propagation delay
- Different latencies in different directions when nodes on the path adopt different sleep schedules
- Skew sleep schedules to allow rapid data forwarding in one direction, and slow forwarding in the opposite direction



Configure nodes on different schedules to get different delay effects

Different latencies in different directions, simulation result from ns-2:

- Topology
  - 11 nodes in a line
- Results:
  - Latency increases linearly with the number of hops on both directions
  - Data transfers quickly in the 200ms direction and slowly in the other direction



Latency Analysis

In a line topology of N nodes (no adaptive listening)

- P: schedule phase difference
- T_f: length of a frame
- $t_{cs,n}$: carrier sense delay at hop n, which is random
- t_{cs}: mean carrier sense delay
- t_{tx}: transmission delay
- $D(N)$: total delay

- $P > t_{cs,n-1} + t_{tx}$ at each Hop n

  $E[D(N)] = T_f/2 + (N - 1)P + t_{cs} + t_{tx}$
- $P < t_{cs,n-1} + t_{tx}$ at each Hop n,

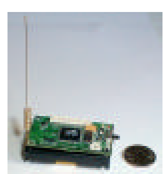
  $E[D(N)] = T_f/2 + (N - 1)(P + T_f) + t_{cs} + t_{tx}$

Conclusions:

- Average latency linearly increases with the number of hops
- Average latency can be controlled by adjusting P

Implementation and Demo

- Simulation: ns-2
- Implementation:
  - Motes running TinyOS
  - PC-104
- Visualization: NAM in real time



PC/104 with moteNIC

Conclusions

- S-MAC can adopt single global schedule
- S-MAC can control schedules to get different latency effects
- We have quantified latency analytically and validated those results experimentally