Title
Schedule and Latency Control in S-MAC

Authors
Yuan Li
Wei Ye
John Heidemann

Publication Date
2003

Center for Embedded Networked Sensing

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

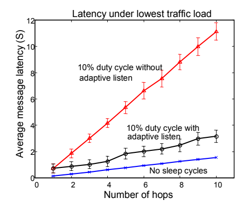
- Medium-access control (MAC) protocol for wireless sensor networks
- Primary goals: *energy conservation* and *self-configuration*
- Low-duty-cycle operation in a multi-hop network
- Nodes form virtual clusters on sleep schedules
- Uses in-channel signaling to avoid overhearing
- Uses Message passing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

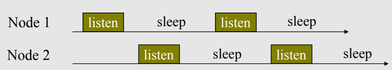


In all three S-MAC modes, latency increases linearly with the number of hops

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:

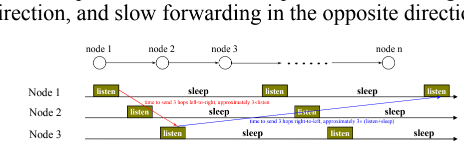
Nodes in multiple clusters can incrementally switch to one global schedule

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

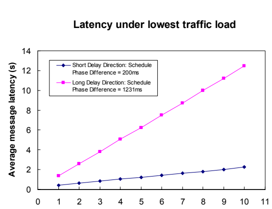
- Select and control sleep schedules to obtain different effects on propagation delay
- Different latencies in different directions when nodes on the path adopt different sleep schedules
- Skew sleep schedules to allow rapid data forwarding in one direction, and slow forwarding in the opposite direction



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