Paper presentation – Ultra-Portable Devices

Paper:

X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks

Michael Buettner, Gary V. Yee, Eric Anderson, Richard Han
University of Colorado

Proceedings of the 4th international conference on Embedded networked sensor systems, 2006, pages 307-320, ACM

Presented by:

Nafiseh Mazloum

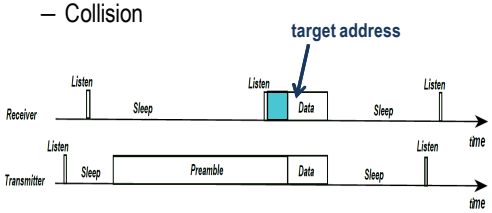
Outline

- Introduction
- X-MAC protocol design
- Adaptation to traffic load
- Experimental evaluation
- Summary

Introduction

- Major sources of energy waste
  - Idle listening
  - Overhearing
  - Control overhead
  - Collision

=> duty cycle: Low power Listening (LPL)

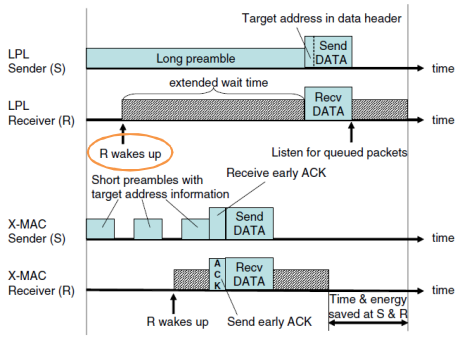


Disadvantages of LPL:

- Long preamble
- Overhearing
- Latency

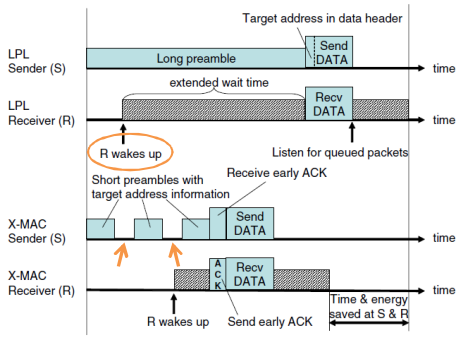
X-MAC Protocol Design

- Embedding target ID in preamble to avoid overhearing



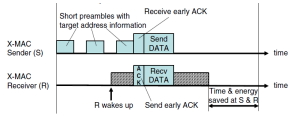
X-MAC Protocol Design (Cont.)

- Reducing long preamble using strobing



Adaptation to Traffic Load Energy Model

- Assumption
  - Packet probability, $P_d(t)$, is known

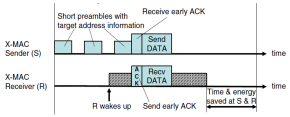


$$E_s = (\text{preamble energy} + \text{energy per ACK listen}) * (\text{expected preamble-listen iterations required}) + (\text{energy to send packet})$$

$$Lat = (\text{duration of preamble} + \text{ACK listen}) * (\text{expected number of iterations required}) + (\text{duration to send packet})$$

Adaptation to Traffic Load Energy Model (Cont.)

- Assumption
  - Packet probability, $P_d(t)$, is known



$$E_r = (\text{listen cycle energy} + \text{sleep cycle energy}) * (\text{expected iterations for a preamble to arrive}) + (\text{energy to send an ACK}) + (\text{energy to receive packet})$$

Adaptation to Traffic Load Optimality

$$E_s = (\text{preamble energy} + \text{energy per ACK listen}) * (\text{expected preamble-listen iterations required}) + (\text{energy to send packet})$$

$$Lat = (\text{duration of preamble} + \text{ACK listen}) * (\text{expected number of iterations required}) + (\text{duration to send packet})$$

Theorem1: energy and latency minimized when *preamble time* and *ACK listen time* set to the lowest value.

Adaptation to Traffic Load Optimality (Cont.)

$$E_r = (\text{listen cycle energy} + \text{sleep cycle energy}) * (\text{expected iterations for a preamble to arrive}) + (\text{energy to send an ACK}) + (\text{energy to receive packet})$$

Theorem 2:

Optimal *Rx-sleep* and *Rx-listen* for min $_{\text{Rx-sleep,Rx-listen}}f(\cdot)$ depends only on $P_d(t)$.

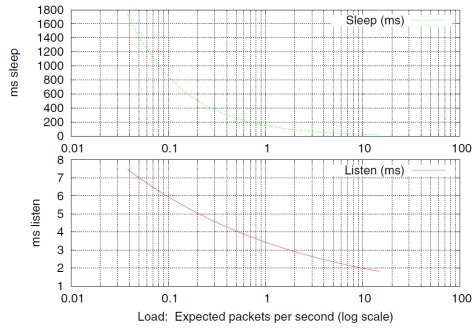
($f(\cdot)$: E_s, E_r, Lat or combination of them)

Theorem 3:

min $_{\text{Rx-sleep,Rx-listen}}f(\cdot)$ can be found by convex optimization tech.

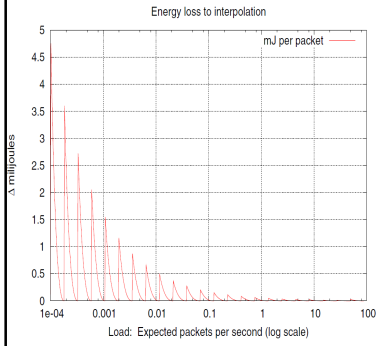
Adaptation to Traffic Load Optimality (Cont.)

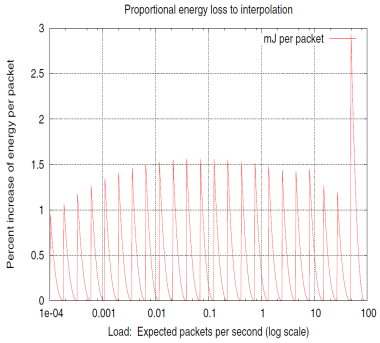
- Motivation for approximation
  - Nonlinear minimization is a too demanding process.
  - Mapping is smooth enough to admit a lightweight approximation.
- Approximation
  - Linear interpolation



Adaptation to Traffic Load Optimality (Cont.)

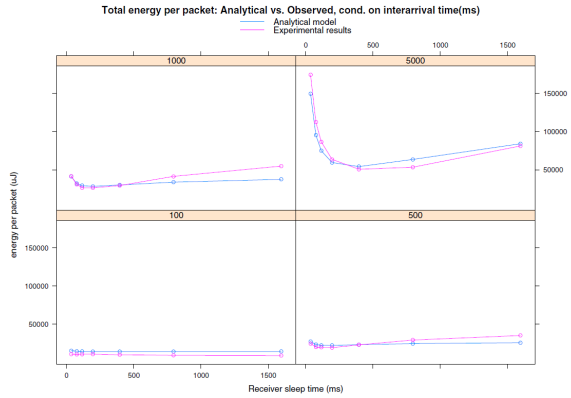
- Energy-efficiency comparison of optimal and interpolated values





Mean difference: 0.45 %
95th percentil defference: 1.3 %

Model Validation

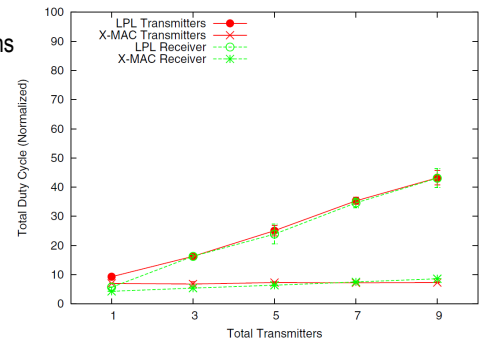


Evaluation and Experimantal Set up

- X-MAC is implemented on top of Mantis OS.
- TelosB with chipcon CC2420 is deployed.
- CC2420 properties
  - 250 kbps
  - 2.4 GHz ISM band
  - Uses 8 MHz TI MSP430 processor with 1 MB external flash
- Processor energy consumprion is ignored.
- Star topology, one RX and up to nine TX.

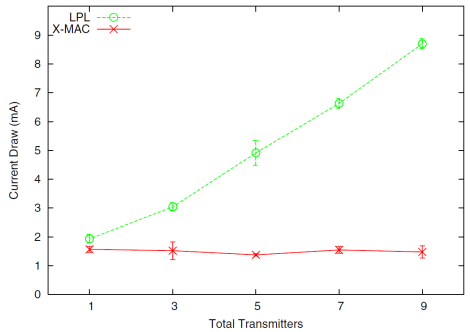
X-MAC Performance

- Duty cycle under no contention
  - Each node send a pkt. every 9 sec.
  - Sleep period 500 ms
  - Preamble 500 ms



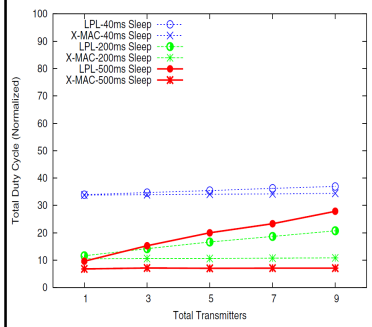
X-MAC Performance (Cont.)

- Energy usage
  - Oscilloscope is attached to one of transmitting node

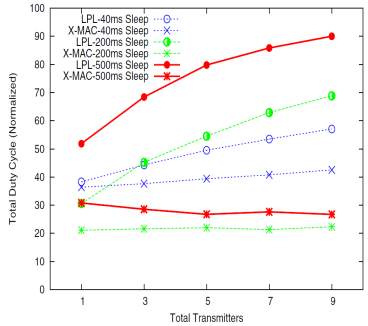


X-MAC Performance (Cont.)

- Duty cycle under contention



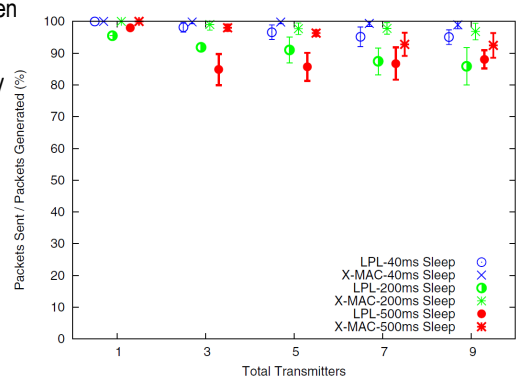
One packet per ten second



One packet per second

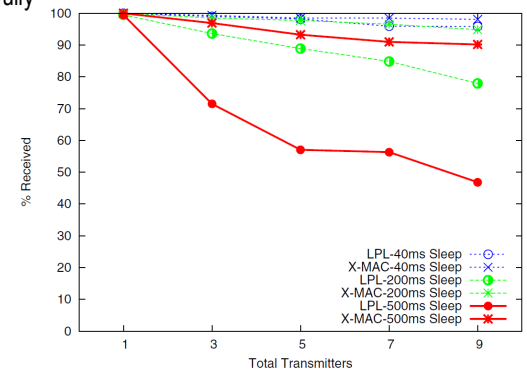
X-MAC Performance (Cont.)

- Fairness
  - Variance between generated pkts and successfully sent pkts



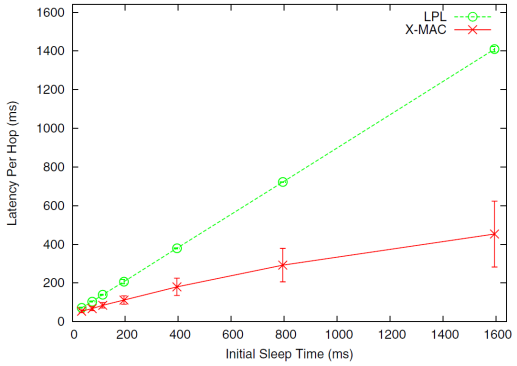
X-MAC Performance (Cont.)

- Transmission success rate
  - No. of successfully received pkts



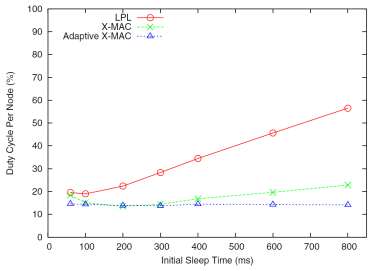
X-MAC Performance (Cont.)

- Latency

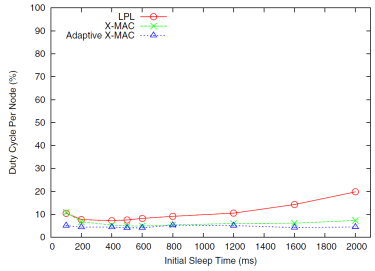


X-MAC Performance (Cont.)

- X-MAC with adaptation
  - Sleep time is adjusted between 220 to 2300 ms
  - Each experiment runs for 10 min



1 packet per ten second



0.1 packet per ten second

Summary

- X-MAC is an approach to avoid overhearing and reduce energy consumption and latency.
- Short preamble and early ACK saves energy and lower the latency.
- Strobed preambles contain target address to avoid overhearing.
- An adaptive algorithm is proposed to select optimal sleep and listen time.