B-MAC

Tunable MAC protocol for wireless networks

Summary of paper "Versatile Low Power Media Access for Wireless Sensor Networks"

Presented by Kyle Heath

Outline

- Introduction to B-MAC
- Design of B-MAC
- B-MAC components
- Evaluation of B-MAC
- Summary

Introduction to B-MAC

- B-MAC = Berkley Media Access Control
- A simple carrier sense media access protocol
  - Link-access protocol only
- Exposes parameters to higher network layers
  - Tunable media access instead of a “black box”

B-MAC Design Objectives

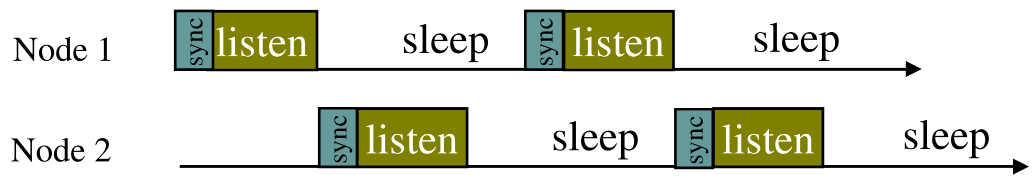
- Principles
  - Reconfigurable MAC protocol
  - Flexible control
  - Hooks for sub-primitives
    - Backoff/Timeouts
    - Duty Cycle
    - Acknowledgements
  - Feedback to higher protocols
  - Minimal implementation
  - Minimal state
- Primary Goals
  - Low Power Operation
  - Effective Collision Avoidance
  - Simple/Predicable Operation
  - Small Code Size
  - Tolerant to Changing RF/Networking Conditions
  - Scalable to Large Number of Nodes

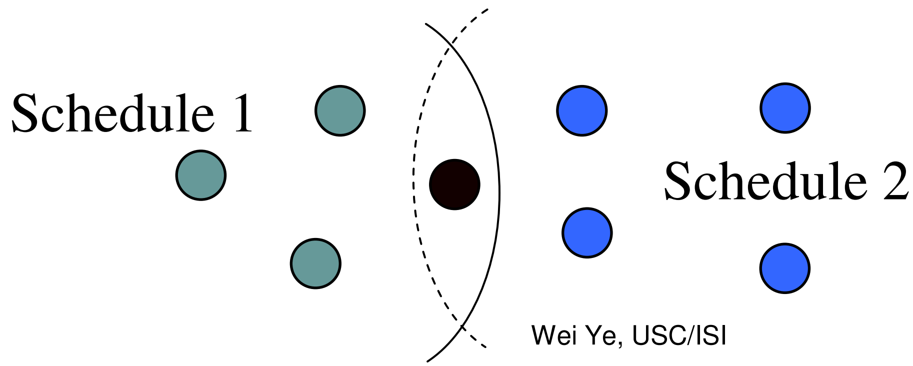
B-MAC Features

- Reconfiguration and control of link layer protocol parameters
  - Acknowledgements, Backoff/Timeouts, Power Management, Hidden Terminal Management (RTS/CTS)
- Ability to choose tradeoffs – “knobs”
  - Fairness, Latency, Energy Consumption, Reliability
- Power consumption estimation through analytical and empirical models
  - Feedback to network protocols
  - Lifetime estimation
- Mechanisms to achieve network protocols’ goals

Other MAC protocols

- **S-MAC**
  Ye, Heidemann, and Estrin, INFOCOM 2002
  - Synchronized protocol with periodic listen periods
  - "Black Box" design
    - Designed for a general set of workloads
    - User sets radio duty cycle
    - SMAC takes care of the rest so you don't have to
    - Integrates higher layer functionality into link protocol





Wei Ye, USC/ISI

- **T-MAC**
  van Dam and Langendoen, Sensys 2003
  - Reduces power consumption by returning to sleep if no traffic is detected at the beginning of a listen period

B-MAC Components

- Channel arbitration
  - Clear Channel Assessment (CCA)
  - back offs
- Reliability
  - Link layer acknowledgements
- Power efficient communication
  - Low Power Listening (LPL)

Note: services like organization, synchronization, and routing are left to higher levels.

Clear Channel Assessment

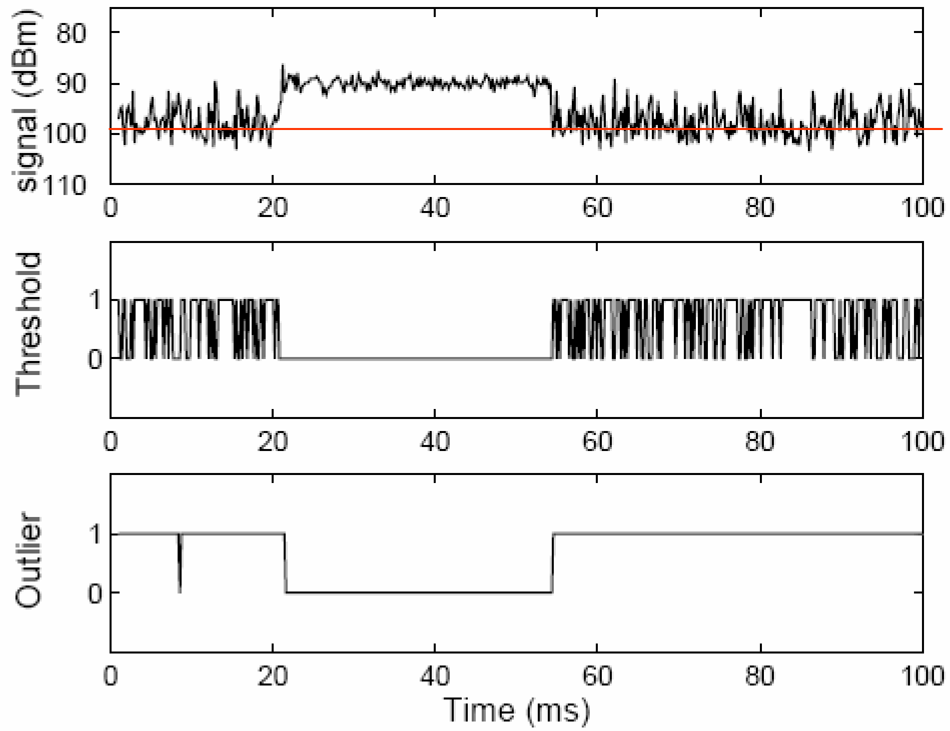


Figure 2: Clear Channel Assessment (CCA) effectiveness for a typical wireless channel. The top graph is a trace of the received signal strength indicator (RSSI) from a CC1000 transceiver. A packet arrives between 22 and 54ms. The middle graph shows the output of a thresholding CCA algorithm. 1 indicates the channel is clear, 0 indicates it is busy. The bottom graph shows the output of an outlier detection algorithm.

Automatic estimation of noise floor

Simple threshold reduces throughput

If no outliers after 5 samples, channel is considered busy

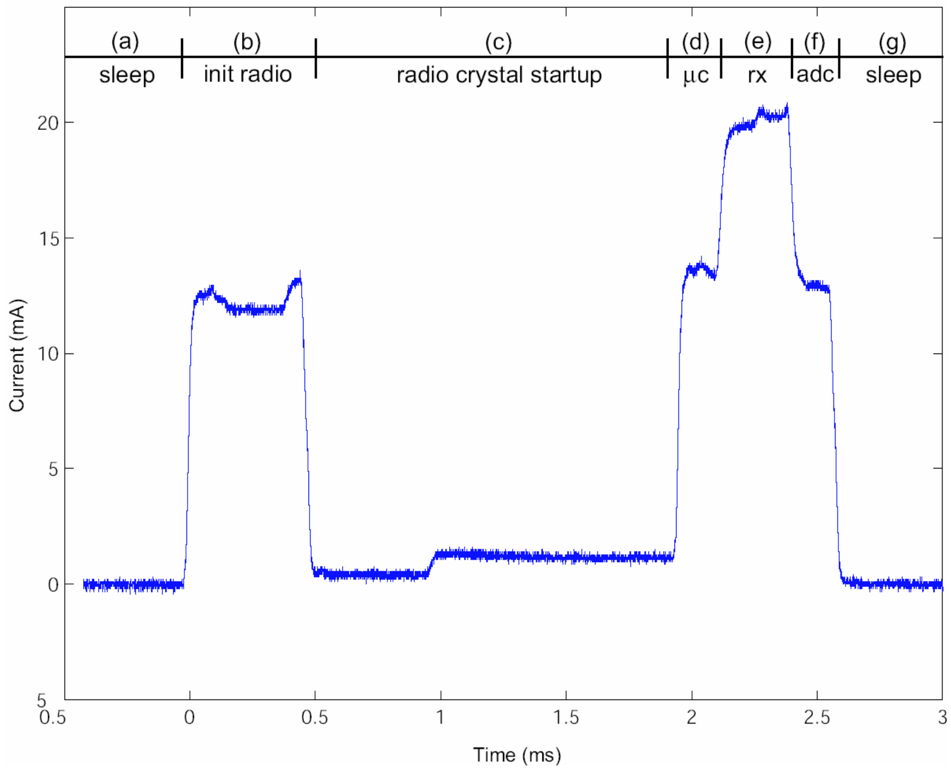
Clear Channel Assessment

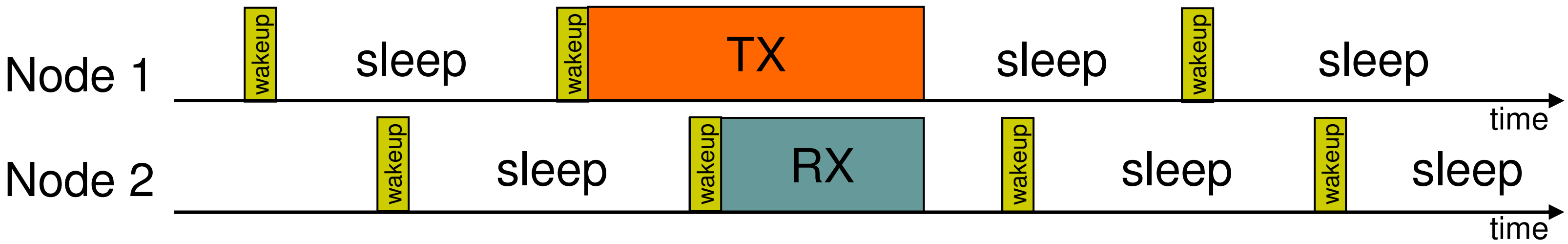
- Configurable “knobs”
  - Enable/Disable CCA
  - Configure initial and congestion back off times

- Adjusts protocol’s
  - Fairness
  - Available throughput

Low Power Listening (LPL)

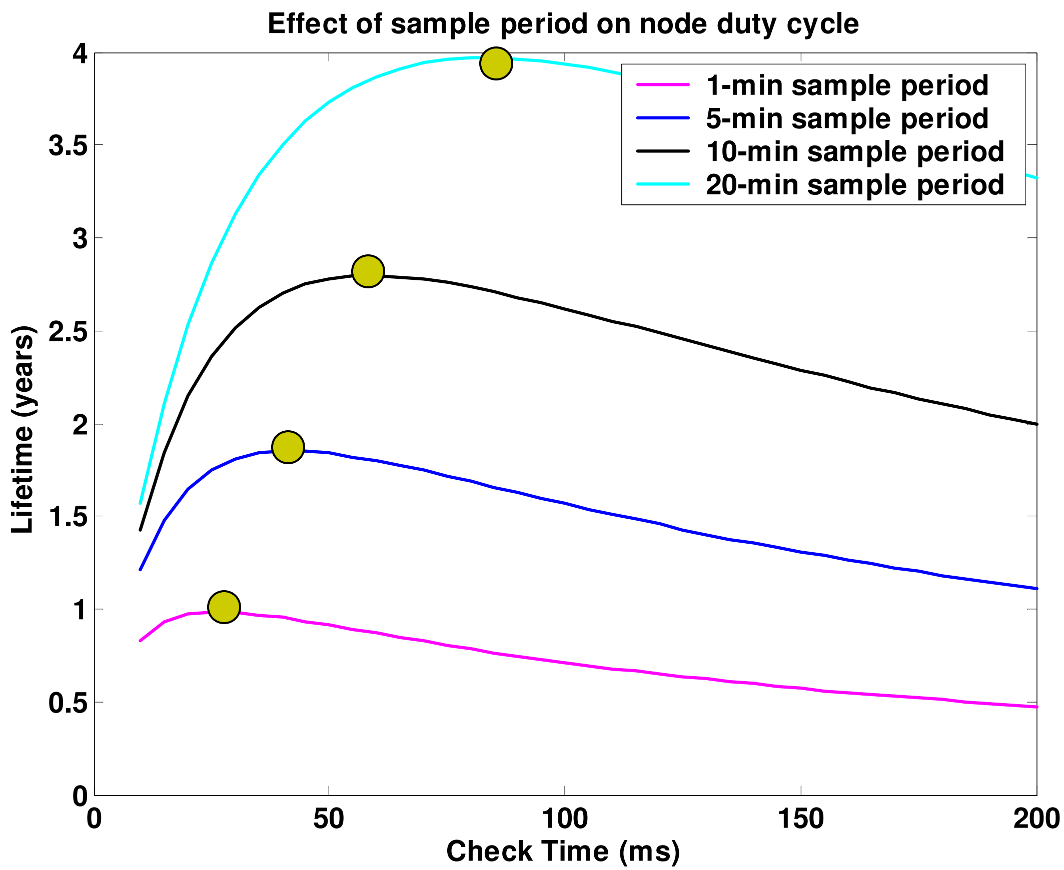
- Higher level communication scheduling
  - Energy Cost = RX + TX + Listen
  - Start by minimizing the listen cost
- Example of a typical low level protocol mechanism
- Periodically
  - wake up, sample channel, sleep
- Properties
  - Wakeup time fixed
  - “Check Time” between wakeups variable
  - Preamble length matches wakeup interval
- Overhear all data packets in cell
  - Duty cycle depends on number of neighbors and cell traffic





Effect of LPL Check Interval

- Single hop data reporting application
- Higher sampling rate
  - Higher traffic in a cell
  - Higher duty cycle
- Optimize the check time to the traffic
  - Application knows sample rate (packet generation rate)



Implementation Size

- Higher level service built on top of B-MAC in order to compare with S-MAC
  - Reliable transport (Acks)
  - Hidden Terminal support (RTS-CTS)
- Implementation smaller than S-MAC

Protocol	ROM	RAM
B-MAC	3046	166
B-MAC w/ ACK	3340	168
B-MAC w/ LPL	4092	170
B-MAC w/ LPL & ACK	4386	172
B-MAC w/ LPL & ACK + RTS-CTS	4616	277
S-MAC	6274	516

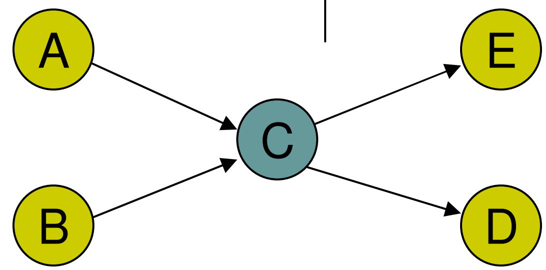
Table 1: A comparison of the size of B-MAC and S-MAC in bytes. Both protocols are implemented in TinyOS.

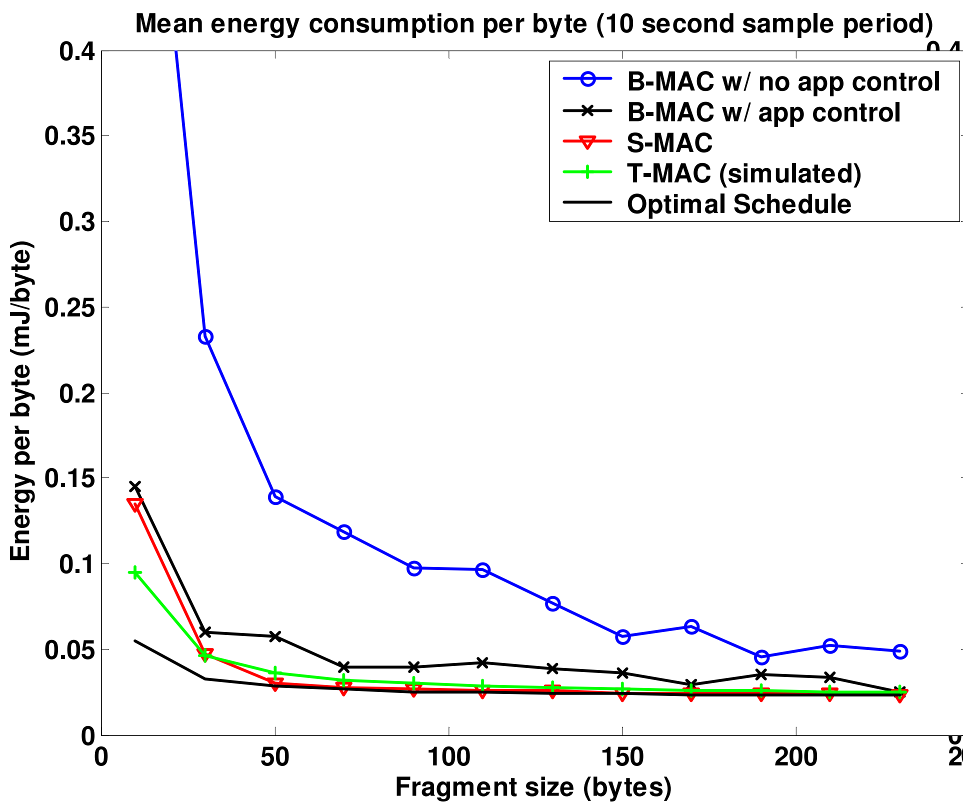
Fragmentation Support

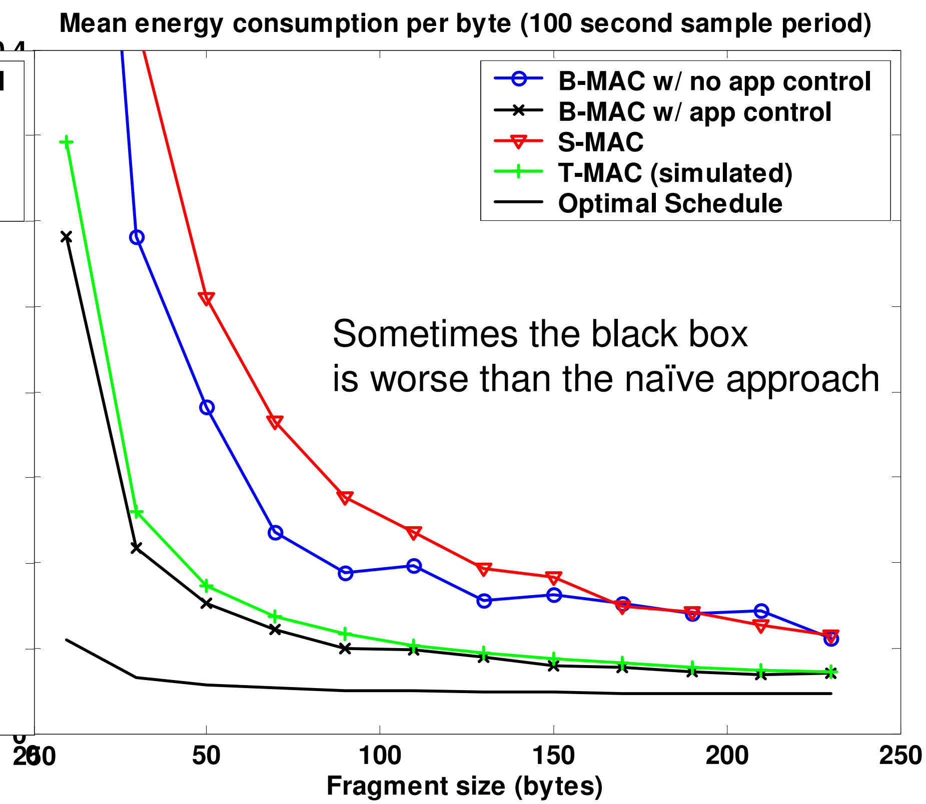
Factored vs Layered Protocol

- S-MAC
  - RTS-CTS Fragmentation Support
- B-MAC
  - Network protocol sends initial data packet with number of fragments pending
  - Disable backoff & LPL for rest of fragments

- Measure energy consumption at C (bottleneck node)
- Minimizing power relies on controlling link layer primitives

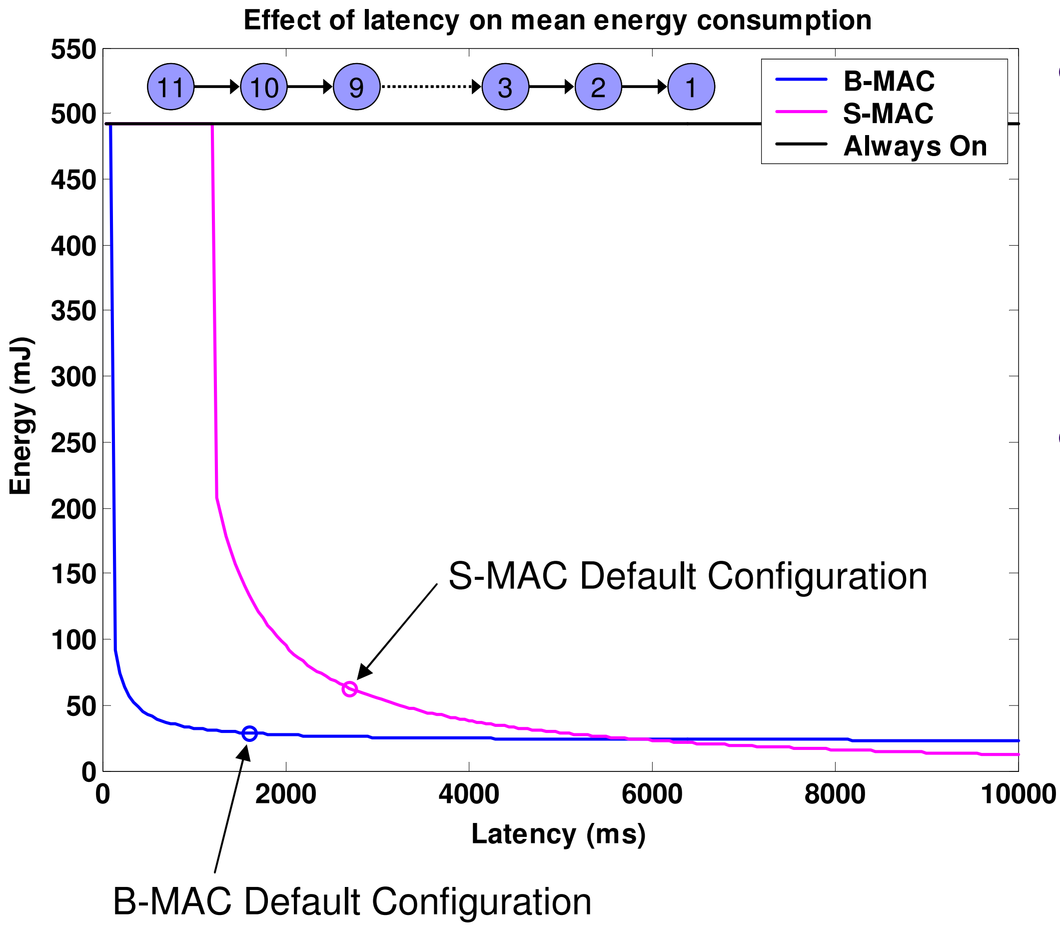






Tradeoffs: Latency for Energy

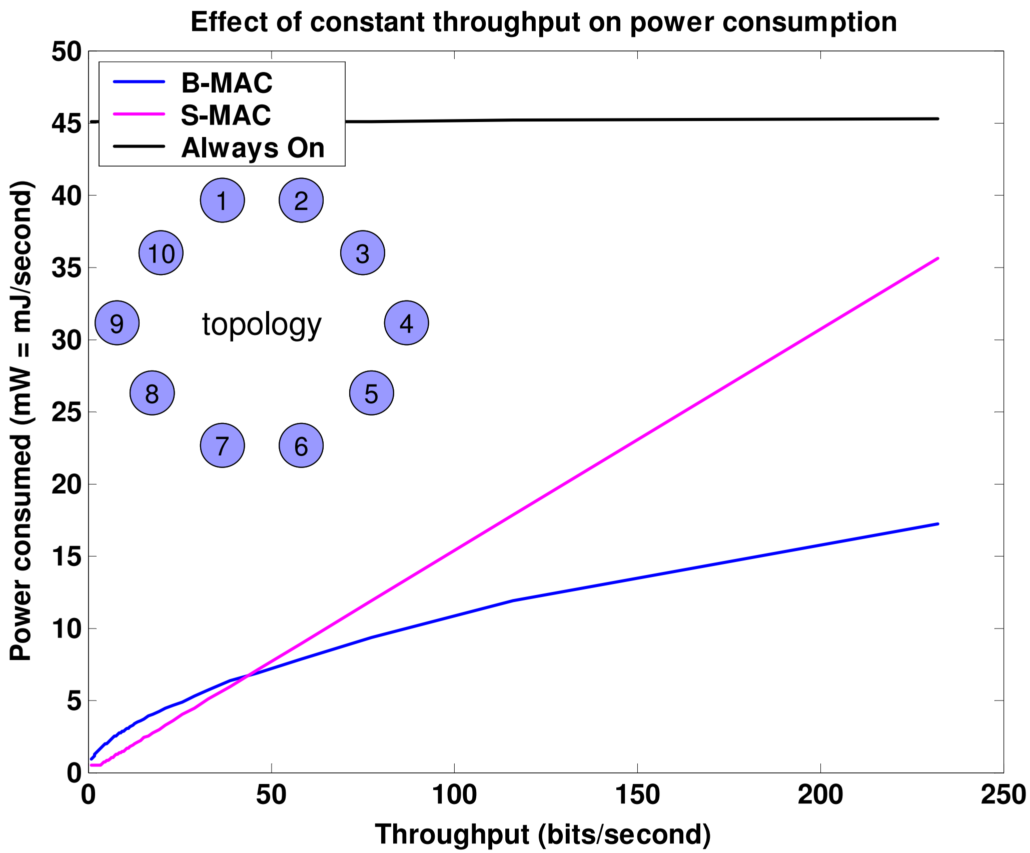
Factored vs Traditional Protocol



- Assume a multihop packet is generated every 10 sec
  - No queuing delay allowed
- Delay the packet
  - S-MAC sleeps longer between listen period
  - B-MAC increases the check interval and preamble length

Tradeoffs: Throughput for Energy
Factored vs Layered Protocol

- 10 node single hop network
  - Increase transmission rate
  - Deliver each packet within 10 sec
  - Measure average power consumption per node
- As throughput increases
  - B-MAC reduces check interval as traffic increases
  - S-MAC uses optimal duty cycle
    - Protocol overhead causes energy to increase linearly



Lifetime Model

$$\min(E) = E_{rx} + E_{tx} + E_{listen} + E_{sleep}$$

- Transmit

$$t_{tx} = r \times (L_{preamble} + L_{packet}) t_{txb}$$

$$E_{tx} = t_{tx} c_{txb} V$$

- Receive

$$t_{rx} = nr \times (L_{preamble} + L_{packet}) t_{rxb}$$

$$E_{rx} = t_{rx} c_{rxb} V$$

Notation	Parameter
r	Sample Rate (packets/sec)
n	Neighborhood size
$L_{preamble}$	Preamble length (bytes)
L_{packet}	Packet length (bytes)
c_{sleep}	Current : Sleep (mA)
c_{rxb}	Current : Rx one byte
c_{txb}	Current : Tx one byte
C_{batt}	Capacity : Battery (mAh)
V	Voltage
t_i	Time : Radio sampling interval (s)
$t_{startup}$	Time : Radio startup
t_{rxb}	Time : Rx one byte
t_{rx}	Time : Rx per second
t_{txb}	Time : Tx one byte
t_{tx}	Time : Tx per second
t_l	Time : Lifetime (s)

Lifetime Model

$$\min(E) = E_{rx} + E_{tx} + E_{listen} + E_{sleep}$$

- LPL Sampling

$$E_{sample} = 17.3\mu J$$

$$E_{listen} \le E_{sample} \times \frac{1}{t_i}$$

- Sleep

$$t_{listen} = t_{startup} \times \frac{1}{t_i}$$

$$t_{sleep} = 1 - t_{rx} - t_{tx} - t_{listen}$$

$$E_{sleep} = t_{sleep} \times c_{sleep}$$

Notation	Parameter
r	Sample Rate (packets/sec)
n	Neighborhood size
$L_{preamble}$	Preamble length (bytes)
L_{packet}	Packet length (bytes)
c_{sleep}	Current : Sleep (mA)
c_{rxb}	Current : Rx one byte
c_{txb}	Current : Tx one byte
C_{batt}	Capacity : Battery (mAh)
V	Voltage
t_i	Time : Radio sampling interval (s)
$t_{startup}$	Time : Radio startup
t_{rxb}	Time : Rx one byte
t_{rx}	Time : Rx per second
t_{txb}	Time : Tx one byte
t_{tx}	Time : Tx per second
t_l	Time : Lifetime (s)

Lifetime Model

$$\min(E) = E_{rx} + E_{tx} + E_{listen} + E_{sleep}$$

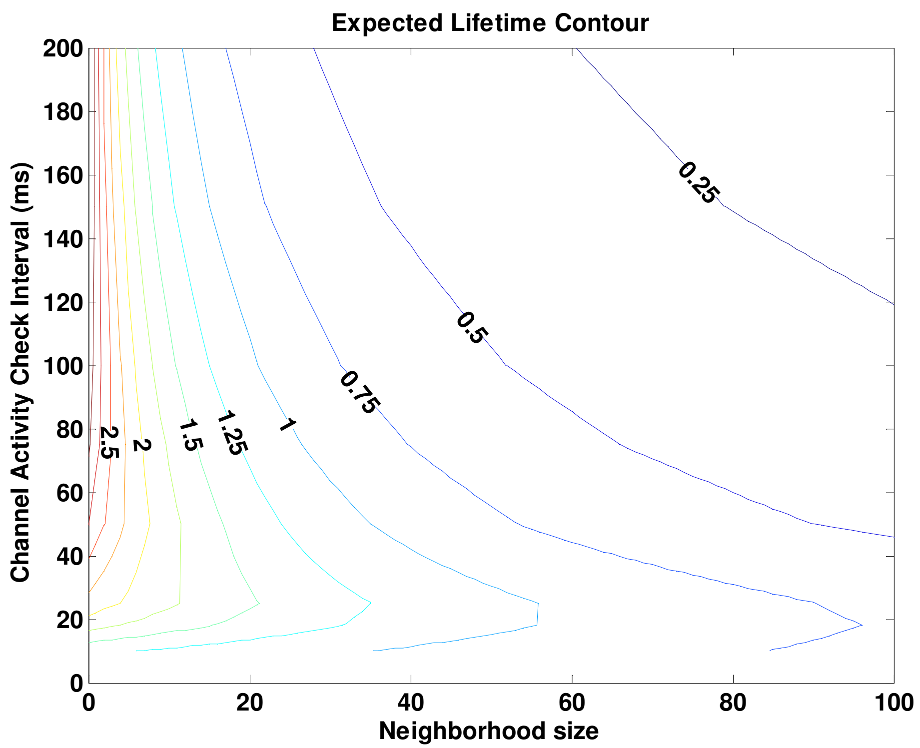
- The total energy, $\boldsymbol{E}$, can be used to calculate the expected lifetime of the system

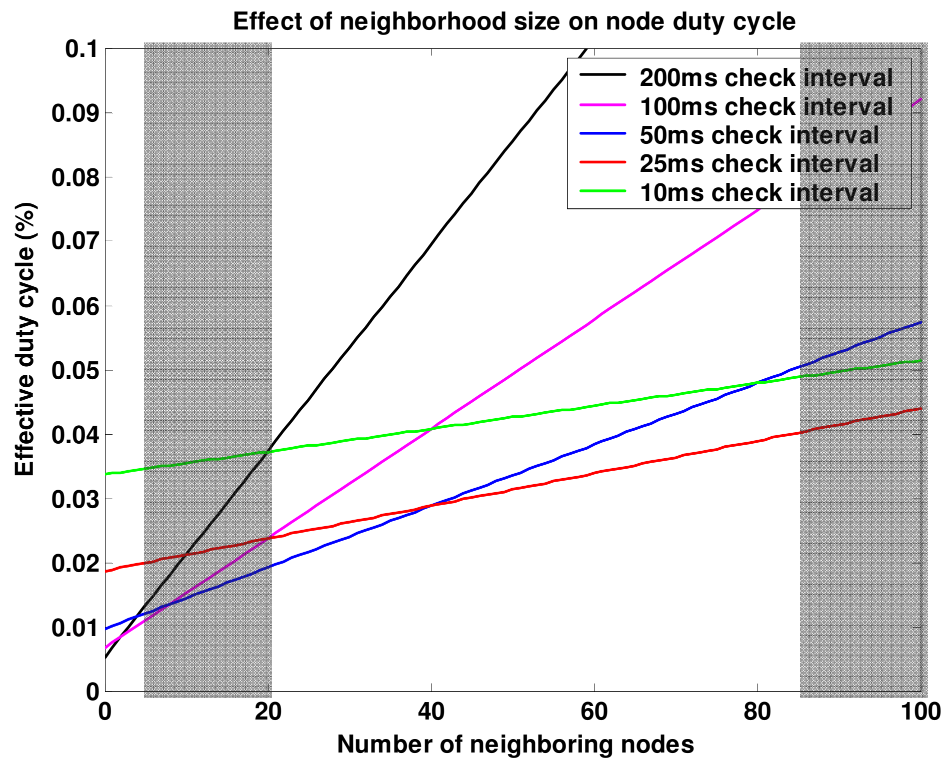
$$t_l = \frac{C_{batt} \times V}{E} \times 60 \times 60$$

Notation	**Parameter**
r	Sample Rate (packets/sec)
n	Neighborhood size
$L_{preamble}$	Preamble length (bytes)
L_{packet}	Packet length (bytes)
c_{sleep}	Current : Sleep (mA)
c_{rxb}	Current : Rx one byte
c_{txb}	Current : Tx one byte
C_{batt}	Capacity : Battery (mAh)
V	Voltage
t_i	Time : Radio sampling interval (s)
$t_{startup}$	Time : Radio startup
t_{rxb}	Time : Rx one byte
t_{rx}	Time : Rx per second
t_{txb}	Time : Tx one byte
t_{tx}	Time : Tx per second
t_l	Time : Lifetime (s)

Effect of Neighborhood Size

- Neighborhood Size affects amount of traffic in a cell
  - Network protocols typically keep track of neighborhood size
  - Bigger Neighborhood → More traffic





Conclusions

- Coordination with higher protocols is essential for long lived operation
- Traditional abstraction at the network layer doesn't fit sensor networks—need a new abstraction at the link layer like B-MAC