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

- **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

---

**Diagram:**

- **Schedule 1**
  - Node 1: sync, listen, sleep, sync, listen, sleep
  - Node 2: sync, listen, sleep, sync, listen, sleep

- **Schedule 2**
  - Wei Ye, USC/ISI
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

Automatic estimation of noise floor

Simple threshold reduces throughput

If no outliers after 5 samples, channel is considered busy

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.
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

<table>
<thead>
<tr>
<th>Protocol</th>
<th>ROM</th>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>B-MAC</td>
<td>3046</td>
<td>166</td>
</tr>
<tr>
<td>B-MAC w/ ACK</td>
<td>3340</td>
<td>168</td>
</tr>
<tr>
<td>B-MAC w/ LPL</td>
<td>4092</td>
<td>170</td>
</tr>
<tr>
<td>B-MAC w/ LPL &amp; ACK</td>
<td>4386</td>
<td>172</td>
</tr>
<tr>
<td>B-MAC w/ LPL &amp; ACK + RTS-CTS</td>
<td>4616</td>
<td>277</td>
</tr>
<tr>
<td>S-MAC</td>
<td>6274</td>
<td>516</td>
</tr>
</tbody>
</table>

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

Sometimes the black box is worse than the naïve approach
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

Effect of latency on mean energy consumption

B-MAC Default Configuration
S-MAC Default Configuration
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**

\[
t_{rx} = n r \times (L_{preamble} + L_{packet}) t_{rxb}
\]

\[
E_{rx} = t_{rx} c_{rxb} V
\]

**Notation**

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

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

- **LPL Sampling**

\[ E_{sample} = 17.3\mu J \]

\[ E_{listen} \leq E_{sample} \times \frac{1}{t_i} \]

- **Sleep**

\[ t_{listen} = t_{startup} \times \frac{1}{t_i} \]

\[ t_{sleep} = 1 - t_{rx} - t_{tx} - t_{listen} \]

\[ E_{sleep} = t_{sleep} \times c_{sleep} \]

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

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

- The total energy, \( E \), can be used to calculate the expected lifetime of the system

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

<table>
<thead>
<tr>
<th>Notation</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r )</td>
<td>Sample Rate (packets/sec)</td>
</tr>
<tr>
<td>( n )</td>
<td>Neighborhood size</td>
</tr>
<tr>
<td>( L_{\text{preamble}} )</td>
<td>Preamble length (bytes)</td>
</tr>
<tr>
<td>( L_{\text{packet}} )</td>
<td>Packet length (bytes)</td>
</tr>
<tr>
<td>( c_{\text{sleep}} )</td>
<td>Current : Sleep (mA)</td>
</tr>
<tr>
<td>( c_{\text{txb}} )</td>
<td>Current : Tx one byte</td>
</tr>
<tr>
<td>( c_{\text{rxb}} )</td>
<td>Current : Rx one byte</td>
</tr>
<tr>
<td>( C_{\text{batt}} )</td>
<td>Capacity : Battery (mAh)</td>
</tr>
<tr>
<td>( V )</td>
<td>Voltage</td>
</tr>
<tr>
<td>( t_j )</td>
<td>Time : Radio sampling interval (s)</td>
</tr>
<tr>
<td>( t_{\text{startup}} )</td>
<td>Time : Radio startup</td>
</tr>
<tr>
<td>( t_{\text{rxb}} )</td>
<td>Time : Rx one byte</td>
</tr>
<tr>
<td>( t_{\text{tx}} )</td>
<td>Time : Rx per second</td>
</tr>
<tr>
<td>( t_{\text{txb}} )</td>
<td>Time : Tx one byte</td>
</tr>
<tr>
<td>( t_{\text{tx}} )</td>
<td>Time : Tx per second</td>
</tr>
<tr>
<td>( t_j )</td>
<td>Time : Lifetime (s)</td>
</tr>
</tbody>
</table>
Effect of Neighborhood Size

- Neighborhood Size affects amount of traffic in a cell
  - Network protocols typically keep track of neighborhood size
  - Bigger Neighborhood $\rightarrow$ 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