Sensor Network Media Access Design

Joseph Polastre
Computer Science Division
EECS Department
University of California, Berkeley
polastre@cs.berkeley.edu

ABSTRACT

Wireless Sensor Networks impose additional challenges on Media Access Control mechanisms. Many schemes have been proposed that use either CSMA or TDMA protocols; however design of a CSMA protocol for WSNs must be more than a general purpose mechanism. In this work, we analyze the parameters of a CSMA layer and their affect on performance. We examine how to use upcalls to notify applications of the underlying channel utilization to enable an application to change its sampling phase, sampling interval, or aggregation methods. To increase performance, an effective carrier sense algorithm is presented for collision avoidance (CA). Using these principles, we implemented a CSMA-CA scheme called B-MAC on the UC Berkeley mica2 platform. B-MAC raises the maximum theoretical bandwidth of the mica2 from 42 packets/sec to 53 packets/sec, outperforms the default 802.11-style TinyOS MAC layer by over 100% (in terms of packet throughput) and achieves 85% channel utilization. B-MAC shows the need for application coordination in wireless sensor network media access layers.

1. INTRODUCTION

Wireless sensor networks (WSNs) have been proposed for many novel applications including microclimate monitoring, distributed control, and tracking. WSNs operate in a new realm of constraints. Nodes are required to run for many months on small power supplies. They must be robust, collect and report data, and use communication to organize.

In both wired and wireless networks, media access has been at the core of effective communication. Many have proposed traditional methods that are similar to 802.11 or Ethernet for media access. Since WSNs are a new domain of wireless application, traditional methods rarely minimize power consumption or provide enough control to the application.

In this paper, we discuss design decisions for wireless sensor network MAC layer design and evaluate the tradeoffs. We implemented a CSMA MAC layer called B-MAC based on the design properties we discuss. We analyze the performance of this MAC layer and show that it can provide significant improvements over current adopted MAC implementations for wireless sensor networks.

2. MAC LAYER DESIGN

In this section, we describe principles about MAC layer design and their tradeoffs. Our analysis is targeted at low power sensor network applications that require the utmost power conservation while relying on a robust layer for transmission.

2.1 Backoff

Of primary concern in achieving high channel utilization in CSMA schemes is the initial and congestion backoff algorithm. The initial backoff is the time a transmitter delays from the point that a packet is submitted for transmission to the first time the status of channel is evaluated. The congestion backoff is the time a transmitter delays after checking the status of a channel and determining that the channel is busy.

The initial backoff affects the maximum load that each node may offer to the channel. Longer initial backoffs result in less throughput per node and more nodes to saturate the channel. Long initial backoffs prevent against collision when responding to synchronized transmissions (such as broadcast messages). Conversely, long backoffs are absolutely horrible in bulk data transfers in wireless sensor networks. Not only must the transmitter and receiver stay on for long periods of time (consuming more power), they may be starved from the channel after they start transmitting or be delayed by fairness metrics. For each type of traffic, the initial backoff may be different. Instead of embedded this backoff into the MAC layer, we advocate a policy manager or application control the initial backoff before submitting a packet for transmission.

Conventional schemes including Ethernet and IEEE 802.11 use an exponentially increasing congestion backoff. In wireless sensor networks, of primary concern is power consumption with a secondary concern of channel fairness. Designing an appropriate congestion backoff algorithm for wireless sensor networks must minimize the amount of time the node is powered. To do this, the algorithm must eliminate the capture effect. In Ethernet networks, Ramakrishnan and Yang propose a binary exponential backoff to solve the capture problem [6]. In order to use non-exponential backoff schemes, we must make sure that the channel is not at capacity and adjust each node's transmission phase such that

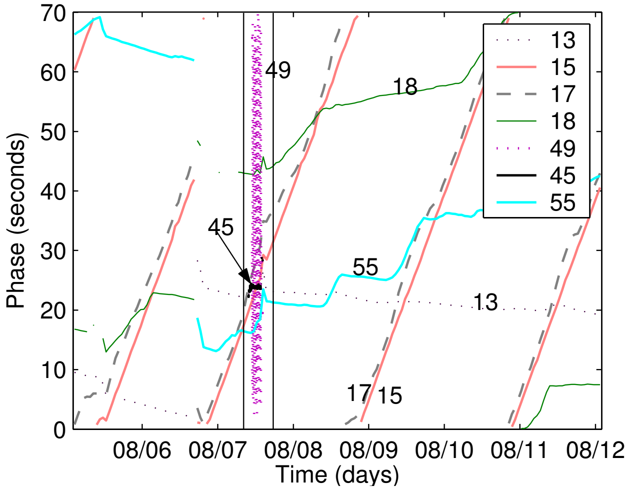

Figure 1: Packet phase of a periodic wireless sensor network application deployed in the Summer of 2002. Each node transmits every 70 seconds, however the application does not change its phase during contention. The result is the capture effect as shown by nodes whose phase is not a horizontal line.

none overlap with others in its cell. When the channel is at capacity, the MAC layer must provide backpressure to the application to delay sending, change period, or change aggregate data. There are two cases to evaluate: adjusting node phase when the channel is not at capacity, and network backpressure when the channel is near capacity.

Phase is only applicable when the available bandwidth in the channel exceeds the current bandwidth usage. In most cases in sensor networks we assume that this is the case. Otherwise, nodes must be awake for long periods of time and we no longer meet our power conservation goals. There are rare situations where channel capacity makes sense–bulk data transfers and node retasking (or reprogramming) are two such events. In those cases, the primary concern is getting a hold of the channel and bulk transferring all of the data such that all nodes can return to sleep as quickly as possible.

Applications may receive notifications that there is contention on the channel. After repeated notifications, logic should be in place to change the phase of sampling or transmission to ease the congestion in the network.

If the channel is near capacity, repeated phase changes will result in contention. The application will know the channel is near capacity due to futile attempts to transmit in different windows. Since we cannot exceed the theoretical maximum bandwidth of the channel, the application must decide how to alter its behavior. During times of channel saturations, all nodes lose in terms of power, fairness, and available bandwidth. A mechanism must be in place to force the application to slow down or disallow packets into the network based on the activity.

2.2 Clear Channel Assessment

Clear Channel Assessment (CCA) is the process of checking the status of the channel and reporting back if there is activity. CCA is required for many different services, including CSMA. CCA allows CSMA schemes to implement collision avoidance (CA). For clear channel assessment to be effective, the algorithm must have a good estimate of the noise floor.

The noise floor on a wireless channel is estimated by examining the characteristics of the channel from the received signal strength indicator (RSSI). The noise floor is typically centered around a mean with some (possibly large) standard deviation. Noise floor estimation is further complicated by errors in the receiver packet detection caused by low-cost radios. Noise floor estimation must be resilient to outliers and noisy signals.

Both CCA and noise floor estimation must be functions that are robust yet quick and require little power to perform. Since they are at the core of CSMA/CA schemes, the power consumed to monitor the channel must be minimized in order to maximize node lifetime. Simpler techniques, such as fixed thresholding, are extremely quick yet have many flaws in channel assessment as seen in section 3. The converse is also true–more complex algorithms take more time (and power), but often are more robust. The challenge is finding an algorithm that achieves both goals.

3. IMPLEMENTATION

In this section, we discuss the implementation of a CSMA MAC layer for wireless sensor networks, called B-MAC, based on the techniques and principles in section 2. As motivation for our CCA and AGC algorithms, we collected 100 traces of RSSI values at 5 kHz from the Chipcon CC1000 [2] FSK radio operating at 433 MHz. We chose the CC1000 radio since it is representative of the wireless sensor network platforms in use today. Mica2 from UC Berkeley [7], Blue from Dust Inc [3], and EmberNet from Ember [4] all use the CC1000 radio.

3.1 Backoff

Since all applications do not have the same requirements on the MAC layer, we chose to expose the underlying events to applications. By default, we chose no initial backoff. This achieves highest channel utilization and allows the application to control the backoff for each packet. For example, bulk data transfers may not want any initial backoff, responses to broadcast messages requires a random backoff, and the backoff after sending a multihop message should be two packet times such that you do not interfere with your parent's retransmission.

On congestion backoff, the application is signaled for a backoff value. If the application chooses not to handle this event, a default handler chooses a random backoff between 0 and 6 ms uniformly.

3.2 Clear Channel Assessment

Clear channel assessment uses a variety of techniques to achieve the performance required for sensor network applications. CCA may be used for additional algorithms including low-power channel sampling for activity. In this mode, the node wakes up, runs the CCA algorithm, and receives data from the channel if it is busy. If the channel is clear, the node returns to sleep. This method has been demonstrated to run at a 1% duty cycle in WSN applications [5].

Our CCA algorithm consists of a free running estimate of the noise floor and a CCA function that is called when the status of the channel is requested. Estimating the noise

```
Data   : size of the FIFO queue, $n$, and exponentially
         weighted moving average weight, $\alpha$
Result: $f$, an estimate of the noise floor
$f \leftarrow$ RSSI;
$q \leftarrow$ new FIFO queue of size $n$;
while radio is on do
    if (end of packet transmission) or (timer fired) then
        if radio is idle then
            $data \leftarrow$ RSSI;
            dequeue($q$);
            enqueue($q$,$data$);
            $m \leftarrow$ median($q$);
            $f \leftarrow \alpha f + (1 - \alpha) m$;
        end
    end
end
```

Algorithm 1: Noise floor estimation

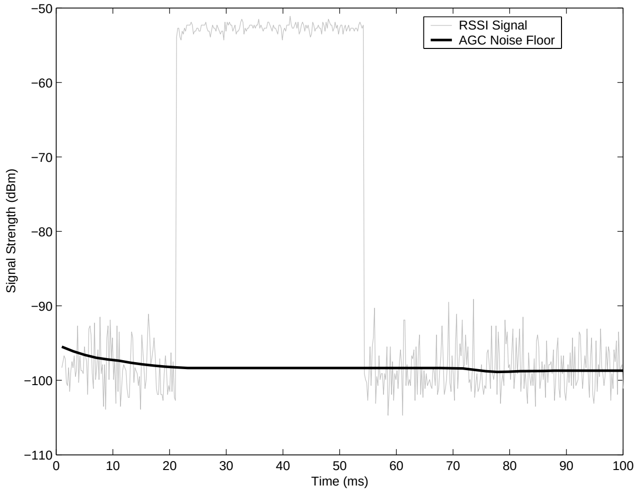

Figure 2: Automatic gain control is critical to accurate noise floor estimation. The estimate of the noise floor is shown with a solid line while the received signal is gray. Notice the estimates are resilient to noise and incoming packets.

floor requires knowing when the state of the radio–one only wants to sample a channel to update the floor if we know the channel is not being used (the radio is in an idle state). However there will certainly be cases where the radio has not yet synchronized to a packet reception, yet we consider the signal value for noise floor estimation. To be resilient to sampling errors and large variances in the noise samples, we developed Algorithm 1.

The premise is as follows: when we think the channel is clear, we sample the signal strength and add it to a FIFO queue. For resiliency, we take the median sample from the queue and update our noise floor estimation using an exponentially weighted moving average with coefficient α. While the radio is on, the noise floor is continually updated for use with the CCA algorithm.

The effectiveness of the noise floor estimation algorithm is shown in Figure 2. In the middle of the figure, one can see a packet arriving and the jump in energy on the channel. The noise floor estimator does take some samples from the beginning of this period; however due to the median selection from the queue, the estimate is not significantly affected. During the remainder of the packet reception, the radio is no longer in an idle state and the noise floor estimate is not updated.

```
Data   : the current estimate of the noise floor, $f$, and the
         number of samples to consider, $s$
Result: true if the channel is clear
for $i \leftarrow 1$ to $s$ do
    $data \leftarrow$ RSSI;
    if $data < f$ then
        return true;
    end
end
return false;
```

Algorithm 2: Clear channel assessment

Figure 3 shows an example RSSI trace with a packet reception. From the signal trace, analysis has shown that noise in the signal when the channel is idle provides some insight on determining the status of the channel. By running the traces through Matlab, we found that detecting the presence of an abnormally low amount of energy on the channel was an extremely good indicator that the channel is clear. The intuition behind this heuristic is that if a node is transmitting on the channel, the probability of that node's signal occurring below the noise floor is extremely unlikely. Even if a transmitter's signal strength is below the noise floor, work by Woo et. al. has shown that packets with such low strength are not reliably receiving and do not make good communication neighbors [10]. Although we have periodic false positives (channel is busy when it is actually clear), this method is significantly more robust than evaluating a single RSSI value or moving averages. The CCA heuristic is shown in Algorithm 2. It is compared to the default CSMA layer in the TinyOS 1.1 release, the TinyOS 1.1 release with adjusted threshold, and the moving average heuristic.

3.3 Other Techniques

Aligned with [10], the preamble length for each message was reduced from 28 bytes in TinyOS 1.1 to 5 bytes in B-MAC. The shorter preamble requires that less bits be used for synchronization in the radio hardware. As a result, nodes whose received signal strength is at or below the noise floor are no longer received by the radio. Our justification for this is two-fold: First, we raise the theoretical limit of the channel from 42 packets/sec to 53 packets/sec. Second, the data in [10, 12] shows that these nodes should not be considered as neighbors for sensor network services.

4. ANALYSIS

The effectiveness of B-MAC described in section 3 is described in this section. We look at the fairness of B-MAC and delivered bandwidth.

4.1 B-MAC Performance

Our first experiment was a periodic sample and send sensor network application. Every 100 ms, a sample is taken from a sensor and sent out over the radio in a 36-byte packet. Each node therefore sends 10 packets/sec and the offered

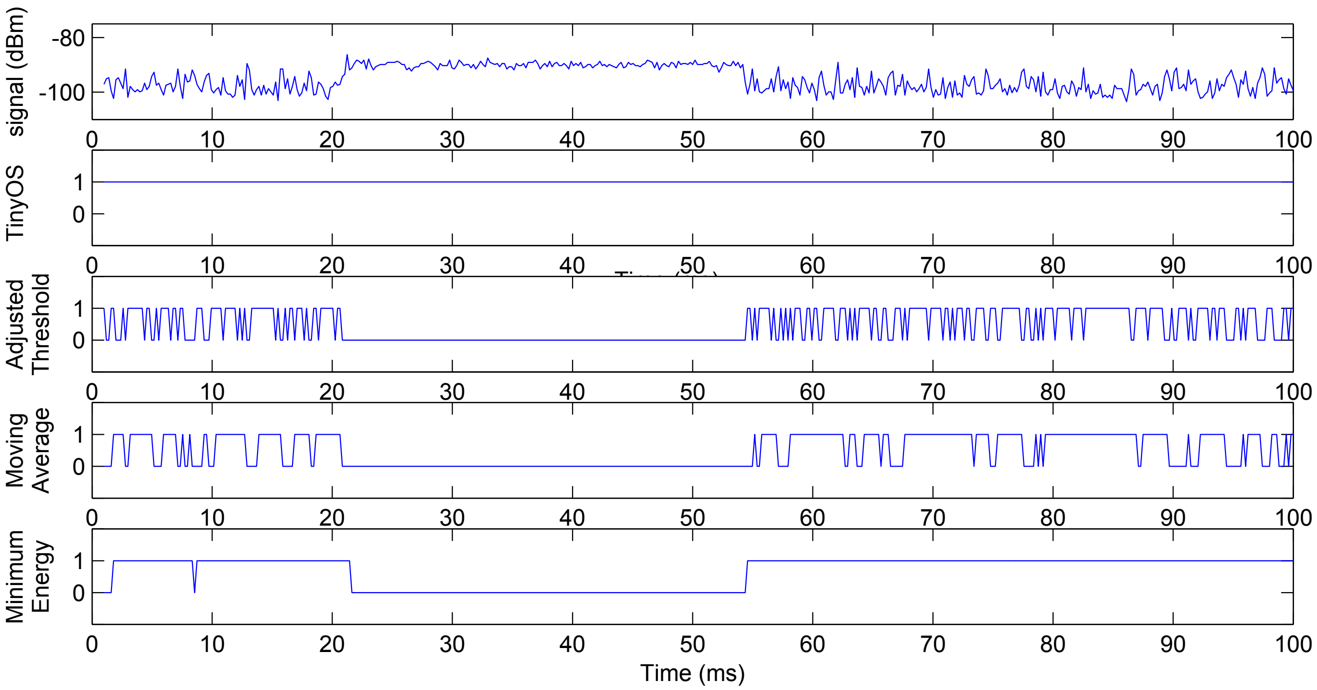

Figure 3: Filtering methods. The top shows the signal received by a Chipcon CC1000 FSK radio. Between 21 and 54ms, the radio is receiving a packet. The channel is idle otherwise idle. The following four graphs show the result of running various clear channel assessment algorithms. A '1' indicates that the channel is clear, '0' indicates the channel is busy. The second graph shows the TinyOS 1.1 clear channel assessment algorithm. The third shows the TinyOS 1.1 algorithm with an adjusted noise threshold. The fourth uses a moving average of 5 samples (1ms) and compares the average with the noise threshold. The last graph shows the result of using the minimum channel energy over 1ms windows to decide if the channel is clear.

load on the channel is $10n$ where n is the number of nodes. The results are shown in Figure 4.

The TinyOS 1.1 MAC layer is the existing adopted MAC that uses a fixed noise floor. In our experiments, we compared TinyOS 1.1 with TinyOS 1.1/AGC and B-MAC. The parameters of each implementation are shown in Table 1

Figure 4 shows the resulting bandwidth from 6 nodes transmitting periodically at 10 packets/sec. All of the nodes are placed in the same radio cell equidistant from each other. Note that automatic gain control assists in higher packet throughput before the channel limit is reached; afterwards its effect is marginal. B-MAC utilizes as much of the channel as possible with no initial backoff. To evaluate the effect of B-MAC after the channel limit is reached, an additional experiment was ran B-MAC with and without a 10-packet queue. We found that queue doesn't help to increase the received bandwidth on the channel, even before the channel reaches capacity.

Up to three nodes, the channel is completely fair and each node achieves 100% packet throughput. At 4 nodes, there is contention in the channel. At 5, the channel becomes saturated, and above 5, the channel has more offered load than the channel can accept. Collected data shows the difference in number of received packets at and beyond channel saturation is at most 8% with no initial backoff.

4.2 Application Controlled Parameters

In this section, we analyze the effect of relaying data to the application about MAC layer events including congestion. We implement a phase-shift scheme to adjust our phase in the event of congestion and analyze the energy usage and fairness of the resulting application.

This is the section that still needs some experiments run...

5. RELATED WORK

Much work has been done on CSMA design for wireless networks. MACAW [1] implements control messages to avoid collisions. They argue that CSMA is too susceptible to the hidden terminal problem. In low bandwidth wireless sensor networks where the contention may not be high, the overhead of control messages is significantly larger than resending the data. Recent work from Whitehouse and Woo [9] has shown that collisions in wireless networks may be detected and the MAC layer may be built with a better understanding of the interference and communication ranges of wireless networks.

In the sensor network space, S-MAC [11] is an energy efficient MAC for sensor networks. Like MACAW, S-MAC relies on control messages to implement collision avoidance. T-MAC [8] provides some improvements over S-MAC in terms of power management, however it is extremely bandwidth limited and the algorithm is no longer applicable after only a fraction of the channel bandwidth is utilized. This makes T-MAC unacceptable for bulk data transfers and reprogramming.

Mote	**TinyOS 1.1**	**TinyOS w/ AGC**	**B-MAC**
Backoff Type	Random (Uniform)	Random (Uniform)	Random(Uniform)
Max Initial Backoff	36+128 bytes 68.3 ms	65 bytes 68.3 ms	0 bytes 0 ms
Max Congestion Backoff	29*16 bytes 193 ms	16 bytes 6.6 ms	16 bytes 6.6 ms
Noise Floor	Fixed	AGC	AGC

Table 1: Comparison of different CSMA MAC layers and their implementation parameters.

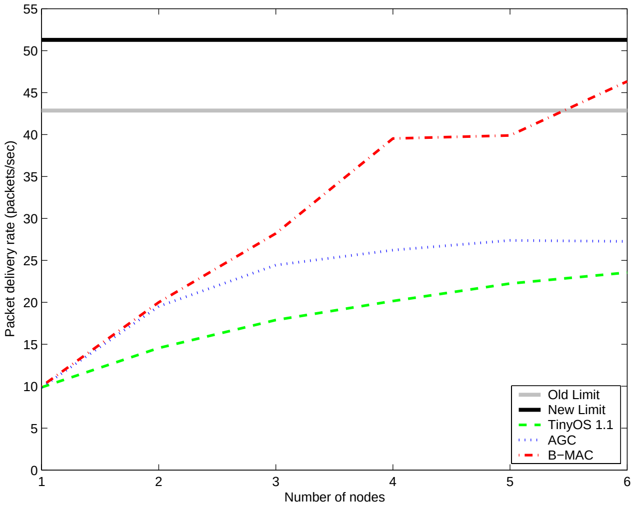

Figure 4: The achievable bandwidth of different MAC layers. Each node is sending data at 10 packets/sec. As the number of nodes increases, the offered load to the channel is increased by 10 packets/sec. The channel has enough offered load at 5 nodes to reach the capacity of the channel.

6. CONCLUSION

We have discussed the domain of wireless sensor networks and the requirements that media access layers must meet. It is not enough to simply impose a uniform MAC layer on the application. Instead, we have shown that the application must have knowledge about the underlying network properties and be able to alter its operation. We demonstrated B-MAC, a CSMA layer for wireless sensor networks, that achieves double the throughput of traditional methods and over 85% channel utilization while being flexible enough to support varying application semantics. Through experience with B-MAC, we advocate application-controlled media access implementations for wireless sensor networks.

7. REFERENCES

[1] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. Macaw: Media access protocol for wireless lans. In *In Proceedings of the ACM SIGCOMM Conference*, 1994.

[2] Chipcon Corporation. CC1000 low power FSK transciever. `http://www.chipcon.com/files/CC1000_Data_Sheet_2_1.pdf`, Apr. 2002.

[3] Dust Inc. Blue. `http://www.dust-inc.com`, 2002.

[4] Ember. EmberNet EM1020. `http://www.ember.com`, 2003.

[5] J. Polastre. Design and implementation of wireless sensor networks for habitat mon itoring. Master's thesis, University of California at Berkeley, 2003.

[6] K. K. Ramakrishnan and H. Yang. The ethernet capture effect: Analysis and solution. In *Proceedings of the IEEE 19th Local Computer Networks Conference*, Oct. 1994.

[7] University of California, Berkeley. Mica2 schematics. `http://webs.cs.berkeley.edu/tos/hardware/design/ORCAD_FILES/MICA2/6310-0306-01ACLEAN.pdf`, Mar. 2003.

[8] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, Nov. 2003.

[9] K. Whitehouse, A. Woo, F. Jiang, and D. Culler. The shadowing phenomenon: implications of receiving during a collision. In *submission*, 2003.

[10] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of multihop routing in sensor networks. In *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, Nov. 2003.

[11] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *In Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, New York, NY, June 2002.

[12] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, Nov. 2003.