The Method of Reducing the Delay Latency to Improve the Efficiency of Power Consumption in Wireless Sensor Networks

Jang Ho, Jeong-Bong Son
Graduate School of Computer Engineering, YeungNam University, Korea

Abstract

Sensor nodes have various energy and computational constraints because of their inexpensive nature and ad-hoc method of deployment. Considerable research has been focused at overcoming these deficiencies through faster media accessing, more energy efficient routing, localization algorithms and system design. Our research attempts to provide a method of improvement MAC performance in these issues. We show that traditional carrier-sense multiple access (CSMA) protocols like IEEE 802.11 do not handle the first constraint adequately, and do not take advantage of the second property, leading to degraded latency and throughput as the network scales in size. We present more efficient method of a medium access for real-time wireless sensor networks. Proposed MAC protocol is a randomized CSMA protocol, but unlike previous legacy protocols, does not use a time-varying contention window from which a node randomly picks a transmission slot. To reduce the latency for the delivery of event reports, it carefully decides a fixed-size contention window, non-uniform probability distribution of transmitting in each slot within the window. We show that it can offer up to several times latency reduction compared to legacy of IEEE 802.11 as the size of the sensor network scales up to 256 nodes using widely used simulator ns-2. We finally show that proposed MAC scheme comes close to meeting bounds on the best latency achievable by a decentralized CSMA-based MAC protocol for real-time wireless sensor networks which is sensitive to latency.

1. INTRODUCTION

In generally wireless networks, every shared channel needs a medium access control protocol to arbitrate access to the channel. For the past decades, many MAC protocols have been designed and several are in operation in wireless networks today. While these protocols work well for traditional data workloads, they are inadequate in emerging wireless sensor networks. This paper discusses the needs for a new look at MAC protocol design in wireless sensor networks, and proposes a new protocol that works well in this problem domain by taking advantage of application requirements and data characteristics.

Several researchers have argued for MAC protocols that conserve energy better than traditional protocols [1] in the sensor network domain. From a different standpoint, our work is motivated by the following reasons for redesigning MAC protocol supports real-time characteristics and is suitable for the sensor networks.

- Most sensor networks are event-driven and have spatially correlated contention.
- Not all sensing nodes need to report an event.
- Time-varying density of sensing nodes.
- Transmission latency is as shortly as possible to satisfy real-time network performance.

These reasons lead to a difference between wireless sensor MAC protocol design and classical MAC design. The goal in our work is to design a MAC protocol that minimizes the time taken to send reports (R) of these messages without collisions in a shared medium where nodes (N) sense an event and contend to transmit on the channel at the same time. Notice that when $R = N$, this becomes the throughput-optimization problem that classical MAC protocols are designed for. When

$R < N$, what we seek is a protocol that allows the first R winners of the contention protocol to send their messages through as quickly as possible, with the remaining $N - R$ potential transmitters suppressing their messages once R have been sent. In the rest of this paper, we denote the number of nodes that have data to send as N, and the number of reports that the sink needs as R.

In traditional CSMA protocols, each node transmits data at a slot picked uniformly at random within the current contention window. This approach does not work well when we are interested in the first R of N potential reports, and has problems scaling well when N suddenly grows. The result is degraded response latency.

Our protocol is proposed on the basis of the concept that when we are interested in low latency for the first R reports, it is important for the first few successful slots to be contention-free. To tightly bound response latency, we use a fixed-size contention window, but a non-uniform, geometrically increasing probability distribution for picking a transmission slot in the window. We give theoretical justification for DPSMAC (differential probability of selection MAC)'s choice of geometrically-increasing probability distribution and show using simulations that DPSMAC can offer up to several times latency reduction as the number of sensors in one radio range scales up to 256 nodes.

We also show that DPSMAC delivers slightly worse throughput than other CSMA protocols when N is small, and slightly better throughput when N is large. Finally, we describe the theoretically optimal non-persistent CSMA MAC when one report of each event is enough, and show that DPSMAC's latency approaches optimal.

2. DPSMAC Design

When a collision occurs, most CSMA protocols specify that the colliding nodes double their value of CW. This is known as binary exponential backoff (BEB). 802.11 [2], B-MAC [3], S-MAC [4], and MACAW [5] are all based on BEB. By increasing CW, most other CSMA protocols attempt to adapt to the current active population size to make a collision-free transmission more likely. There are two problems with this method. First, it takes time for CW to increase to the right value when the active population (N) becomes large, such as when an event is observed by many sensors after a previously-idle period. Second, if CW is already large (because of traffic congestion that has just subsided) and N is small, then such protocols waste bandwidth "backing off." Furthermore, CW is usually chosen to ensure that all active nodes get a chance to send their data, whereas we are interested in the collision-free transmission of the first R of N potential reports of some event. In contrast to previous protocols, DPSMAC uses a small and fixed CW. Of course, nodes can no longer pick contention slots from a uniform distribution, because this would lead to collisions for even moderate values of N. The key difference between DPSMAC and previous CSMA-based wireless MAC protocols is that the probability of picking a slot in this interval is not uniform. Instead, with a carefully-chosen fixed CW and fixed probability distribution, we will show that DPSMAC can perform well in a sensor network. The following intuition leads us to propose the geometrically-increasing probability distribution for picking a contention slot, shown in Figure 1.

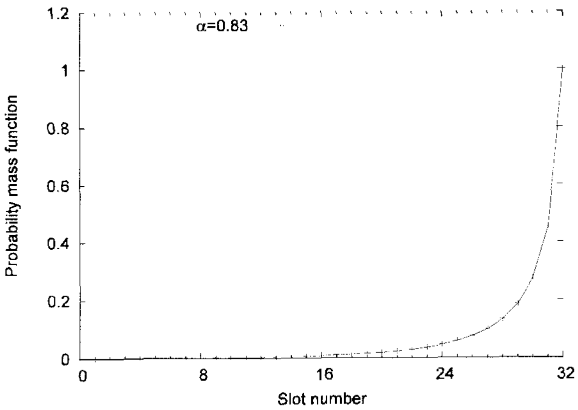

Figure 1. The Probability distribution for the contention slot number.

When N is large, most nodes will choose medium to high slot numbers to transmit, but a small number will choose low slot numbers, making a collision-free transmission likely in a low slot number. When N is medium, most nodes will choose high-numbered slots, making a collision-free transmission likely in a medium slot number. Finally, when N is small, a collision-free transmission is likely in a high slot number 1. Thus, for any N , and no matter how fast N changes, a collision-free transmission is likely. We make this intuition precise in Section 2.1.

2.1 The DPSMAC Probability Distribution

Suppose each sensor picks a slot $r \in [1, CW]$ with probability P_r. We say that slot r is silent if no sensor chooses that slot, and there is a collision if more than one sensor chooses that slot. Also, a sensor wins in slot r if it is the only one to choose slot r, and all others choose later slots. Finally, there is success if some sensor wins some slot in $[1, CW]$.

We use the truncated, fast increasing geometric distribution

$$P_r = \frac{(1-\alpha)\alpha^{(CW-r)}}{1-\alpha^{(CW-r+1)}} \quad \text{for} \quad r = 1, \ldots, CW, (1)$$

where $0 < \alpha < 1$ is a distribution parameter. For these values of α , P_r increases exponentially with r, so the later slots have higher probability.

To motivate this choice, view each sensor's choice of which slot to pick as a decision procedure with CW stages. Each node starts in stage 1 with some overestimate N_1 of N and chooses slot 1 with a small probability. If no sensor chooses slot 1, that is an indication that N_1 is an overestimate of N , so each node updates its guess of the population size by decreasing N_1 to N_2, and proceeds to choose slot 2 with a different probability in stage 2. If slot 2 is also silent, this guess is reduced to N_3 in stage 3, and so on; in general, N_r is the updated guess after there is silence in slots 1, . . . , $r-1$. We have shown that a near-optimal choice of α for a wide range of population sizes is $\alpha = N_1^{-\frac{1}{CW-1}}$ from $N_{CW} = N_1\alpha^{CW-1} = 1$.

We had carried out experiment in which N sensors choose slots using the distribution in Equation 1 with $\alpha = 256^{-(1/(32-1))} \approx 0.836$. Our results verified that we had chosen the correct α , and that over this range, the success rate is not sensitive to the exact choice of α . although the sensors do not know N and use a fixed distribution P_r , the probability of a successful transmission is constantly high for a large range of N .

2.2 Exploring the CSMA Design Space

Current sensor network designs (such as B-MAC [3], the MAC layer of TinyOS) use a fixed-window CSMA protocol, choosing contention slots uniformly at random. The ad-vantage of this design choice is simplicity, and good performance under most practical sensor network deployment scenarios. The disadvantage of this design choice is a lack of scalability under highly correlated traffic or large numbers of sensor nodes.

Bharghavan *et al.* proposed MACAW [5], a MAC protocol for wireless local area networks. MACAW uses BEB, and so without some way to share information about the state of the wireless medium, MACAW would suffer from the well-known Ethernet capture problem: a station that just transmitted resets its contention window to the minimum value, and is thus more likely to transmit again in subsequent competitions. MACAW's solution to this belongs to a class of techniques that we term shared learning. Stations copy the CW value of a station that transmits to their own CW value, and modify BEB so that instead of resetting CW after a successful transmission, decreases it linearly (a multiplicative increase, linear decrease policy).

Instead of shared learning, 802.11 [2] uses memory to solve the fairness problem. When stations begin to compete, they set a countdown timer to a random value picked uniformly from the contention window CW . When the medium becomes busy, the station pauses the countdown timer. When the medium becomes idle and the station wants to compete again, 802.11 resumes its countdown timer. When the countdown timer expires, the station begins its transmission.

In 802.11, a station that successfully transmits resets its CW to a small, fixed minimum value of CW. Consequently, the station has to rediscover the correct CW, wasting some bandwidth.

Table 1 summarizes the design parameters we have carried out a survey. We now show that this particular point in the design space results in good performance in sensor networks with respect to latency, throughput, and fairness.

Table 1. design parameters in the contention window-based CSMA space.

Protocol	Contention window	Shared learning	Memory	Distribution function
802.11	variable	No	yes	Uniform
MACAW	variable	Yes	no	Uniform
S-MAC	Fixed	No	no	Uniform
Proposed MAC	fixed	No	no	geometric exponential

3. PERFORMANCE EVALUATION

In our experiments, we compare DPSMAC configured with CW=32 and α = 0.836 to 802.11. We choose the 802.11 family because it is a practical CSMA protocol whose mechanism for adapting to the number of transmitting stations (BEB) has been included, unmodified, in several proposals for the MAC layer of a sensor network [4, 6].

We run experiments using version 2.31 of the ns-2 [7] network simulator, with all nodes within range of a common base station. We modify all the MACs in our experiments to perform suppression: if a sensor hears R acknowledgments for motion event E from the base station, it suppresses its report of E and removes E's packet from its transmit queue. For experiments with small data packets (40 bytes), we compare DPSMAC, 802.11 without RTS/CTS. For the fairness experiments in Section 3.3, where data packets are 1500 bytes long, we enable RTS/CTS for both DPSMAC and the 802.11 protocols. All experimental results average 20 runs using different random seeds for each run, except the fairness experiments in Section 3.3 which average 5 runs.

3.1 Latency Experiments

We begin by evaluating latency under the constant-rate event workload. To capture varying propagation delays in the environment, variations between sensor electronics, and uncertainty in software system delays on the sensor nodes themselves, we add a random delay in [0, 1] ms to the time that each sensor sends its event report. We measure the time for the base station to receive the first(bottom of error bar), median(point in error bar), and 90th percentile(top of error bar) event report. We plot these times as a function of N. Figure 2 shows the results of this experiment. When N is small, the minimum 802.11 contention window size is large enough to quickly resolve contention between the nodes.

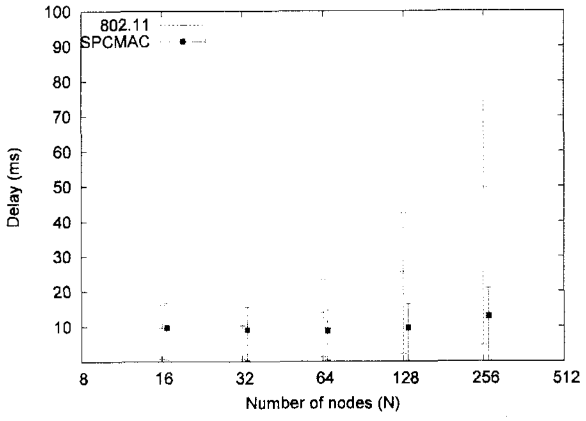

Figure 2. Latency as a function of N (number of sensors reporting an event).

As N grows, however, the 802.11 contention window needs to grow before even one event report can be successfully sent (see the bottom of the error bars in Figure 2), while DPSMAC's fixed contention window resolves contention in constant time with respect to N . Turning to the median and 90th percentile event reporting times, we see that DPSMAC also improves latency for these measures as well, up to N =256. This is primarily due to DPSMAC's improvement in the first event reporting time, but it also shows that DPSMAC can deliver enough throughput to keep up with 802.11 as it sends subsequent event reports.

3. 2Throughput Experiments

We now compare the throughput of DPSMAC, 802.11 under a variety of workloads that saturate the capacity of the wireless medium.

In this experiment, we measure DPSMAC's raw, steady-state throughput on a non-event-driven workload. The purpose of these experiments is to show that even though DPSMAC performs extremely well under an event-based workload, it does not sacrifice much steady-state throughput in an ad-hoc network setting where there are some number of almost constant-bit-rate flows operating concurrently. 32 CBR flows compete to send as much data as possible, using DPSMAC and 802.11. The loss in throughput happens not because CBR flows incur collisions with DPSMAC, but because the winning station wins in a late slot in the case of DPSMAC. The winning 802.11 flow, however, wins in an earlier slot. DPSMAC thus incurs several contention backoff slot delays per transmission, unlike 802.11. When the number increases further, DPSMAC actually outperforms 802.11 in terms of raw throughput (Figure 3). This is because DPSMAC does not incur many collisions, and when N increases, the shape of the DPSMAC distribution makes the winning slot number decrease. This results in less wasted aggregate bandwidth.

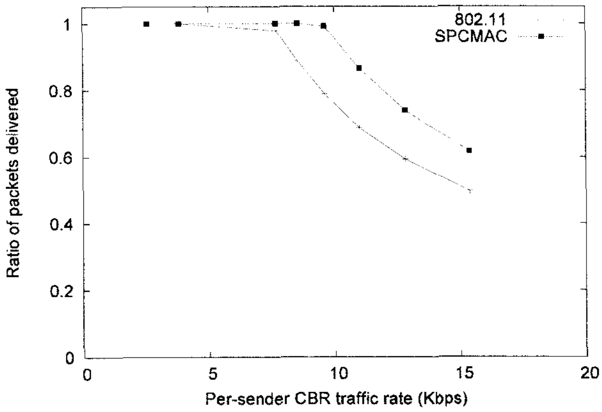

Figure 3. Packet delivery ratio as a function of per-sender CBR traffic rate(32 traffic sources).

3.3 Fairness Experiments

We now examine whether DPSMAC fairly allocates bandwidth between stations. It has been shown that 802.11 does not, but that minor changes to 802.11 can yield a fair protocol [8]. We duplicate the experimental setup given by the authors of the distributed fair scheduling (DFS) protocol. We place some even number of nodes in the same radio range, and set up a traffic pattern where each node is either a traffic source or a traffic sink. The packet size is 1500 bytes, and the RTS/CTS exchange is enabled for both 802.11 and DPSMAC. We ensure that each node is backlogged so that the offered load exceeds the available wireless capacity. Figure 4 shows the throughput achieved by each node in six seconds as a function of the node number. Note that as expected, DPSMAC outperforms 802.11 in terms of fairness. DPSMAC does not in fact achieve a perfectly-fair bandwidth allocation. We expect that this is not a major issue, since sensor networks will contain many redundant nodes reporting similar observations about the environment. However, due to the simplicity of DPSMAC, we expect that a similar approach to DFS could be applied to DPSMAC should fairness become an issue.

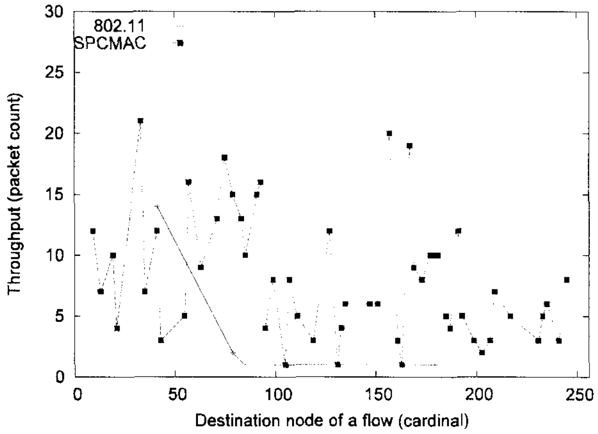

Figure 4. Fairness comparison of 802.11 and DPSMAC.

4. CONCLUSION

We have presented DPSMAC, a MAC protocol for wireless sensor networks that performs well when spatially-correlated contention occurs and adapts well to sudden changes in the number of sensors that are trying to send data. DPSMAC is ideal for sensor networks, where it is often sufficient that any R of N sensors that observe an event report it, with the remaining nodes suppressing their transmissions. The main idea in DPSMAC is to use a geometrically-increasing probability distribution within a fixed-size contention window, rather than varying the window size as in many traditional MAC protocols. Using event-driven experiments, we have shown that DPSMAC outperforms 802.11 and other BEB-based protocols both when the ratio R/N is low, and when both N and R are large.

5. References

[1] Cai, Z., Lu, M., Wang, X., Randomized Broadcast Channel Access Algorithms for Ad Hoc Networks, Proc. of the IEEE Intl. Conf. on Parallel Processing, (2002) 151–158.

[2] IEEE 802.11 Standard, Wireless LAN Medium Access Control and Physical Layer Specifications (1999).

[3] Polastre, J., Hill, J., Culler, D., Versatile Low Power Media Access for Wireless Sensor Networks, Proc. Of the ACM SenSys Conf., Baltimore, MD (2004) 95–107.

[4] Ye, W., Heidemann, J., Estrin, D., An Energy-Efficient MAC Protocol for Wireless Sensor Networks, Proc. of the IEEE INFOCOM Conf., New York, NY (2002) 1567–1576.

[5] Bharghavan, V., MACAW: A Media Access Protocol for Wireless LANs, Proc. of the ACM SIGCOMM Conf., London, UK (1994) 212–225.

[6] Woo, A., Culler, D., A Transmission Control Scheme for Media Access in Sensor Networks, Proc. of the ACM MOBICOM Conf., Rome, Italy (2001) 221–235.

[7] USC ISI, ns-2 Notes and Documentation (2007), available at <http://www.isi.edu/nsnam/ns>

[8] Vaidya, N., Bahl, V., Gupta, S., Distributed Fair Scheduling in a Wireless LAN, Proc. Of the ACM MOBICOM Conf., Boston, MA (2000) 167–178.