Medium Access Control with Dynamic Frame Length in Wireless Sensor Networks

Dae-Suk Yoo* and Seung Sik Choi*

Abstract—Wireless sensor networks consist of sensor nodes which are expected to be battery-powered and are hard to replace or recharge. Thus, reducing the energy consumption of sensor nodes is an important design consideration in wireless sensor networks. For the implementation of an energy-efficient MAC protocol, a Sensor-MAC based on the IEEE 802.11 protocol, which has energy efficient scheduling, has been proposed. In this paper, we propose a Dynamic S-MAC that adapts dynamically to the network-traffic state. The dynamic S-MAC protocol improves the energy consumption of the S-MAC by changing the frame length according to the network-traffic state. Using an NS-2 Simulator, we compare the performance of the Dynamic S-MAC with that of the S-MAC protocol.

1. INTRODUCTION

Wireless Sensor Networks (WSNs) are an emerging technology that has a wide range of potential applications including environmental monitoring, smart spaces, medical systems and robotic exploration. Such networks will consist of large numbers of distributed nodes that organize themselves into multi-hop wireless networks. Each node has one or more sensors, embedded processors and low-power radios and is normally battery operated. Typically in WSNs, the nodes coordinate locally to perform data processing and deliver messages to a common sink node. The important design features for medium access control protocols in a WSN are:

- **Energy** : Each node is normally battery operated. Energy efficiency is a critical issue in order to prolong the network lifetime, because it is often not feasible to replace or recharge batteries for sensor nodes. In particular, MAC protocols must minimize the radio energy costs in sensor nodes.
- **Latency** : The latency requirements depend on the applications or network systems. In the case of a network monitoring system, an event detected needs to be reported to a sink node in real time, so that appropriate action can be taken promptly.
- **Throughput** : The throughput requirements also vary with the application or network system. For example, applications for measuring the variation of temperature need to be designed so that the sink node receives the messages from the nodes periodically. In other

systems, such as fire detection systems, it may suffice for a single report to arrive at the sink.

• **Scalability** : Considering that sensors are movable, sensor networks must allow for scalability in the sense that nodes may be added to the network or removed if their battery is entirely consumed.

Among these important requirements for MACs, energy efficiency is typically the primary goal in WSNs. Previous works involving the IEEE 802.11 protocol identified idle listening as a major source of energy wastage [1]. As the traffic load in many sensor network applications is very light most of the time, it is often desirable to turn off the radio when a node is not participating in any data delivery. Therefore, S-MAC [2] provides a tunable periodic active/sleep cycle for sensor nodes. During the sleep periods, the nodes turn off their radio to conserve energy. During the active periods, the nodes turn on their radio to Tx/Rx messages. This increases energy efficiency, thereby improving the sensor network's lifetime. However, it still has the shortcoming of the trade-off between throughput and latency. Therefore, a T-MAC that has a flexible active interval was introduced for the MAC protocol in order to control the frame size dynamically [3]. In section 2 of this paper, we compare the various MAC protocols and analyze the essential components of an energy efficient MAC protocol. In section 3, we propose the dynamic S-MAC protocol, which is more efficient and flexible with regard to network traffic states. In the conclusion in section 4, we evaluate the performance of the dynamic S-MAC protocol in comparison with that of the S-MAC protocol.

2. RELATED WORK

There are several solutions to the problem of energy wastage due to idle listening. In general, some kind of duty cycle is involved, with each node having active/sleep cycles. For example, TDMA-based protocols are naturally energy preserving, because they have a duty cycle built-in, and do not suffer from collisions [4]. However, maintaining a TDMA schedule in ad-hoc networks is not an easy task and requires much complexity in the nodes. Keeping a list of the neighbor's schedules takes valuable memory capacity. Allocating TDMA slots is a complex problem that requires coordination. Furthermore, as TDMA divides time into very small slots, the effect of clock drift can be disastrous; exact timing is critical.

Another method of energy conservation is to use an extra radio, the so-called 'wake-up radio', which operates on a different frequency than the radio used for communication [5]. As the 'wake-up radio' is only for waking up other nodes, it needs no data processing and therefore uses much less energy. However, it requires an extra component on the node and it doesn't have a positive effect on energy efficiency, because the wake-up radio consumes energy constantly. Therefore, most wireless sensor nodes currently used in research only have a single radio that operates on a single frequency.

Introducing a duty cycle into a contention-based protocol that only uses a single frequency requires some kind of in-band signaling. The well-known CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) protocol, for example, tackles the idle-listening problem. The CSMA protocol is a type of protocol that attempts to avoid collisions. In this protocol, each node overhears its neighboring node's control packets and doesn't access other nodes for using this control

packet. In this way, unnecessary energy consumption can be avoided. However, this function requires that the radio channel be turned on to receive a signal without knowing when it will be transmitted. As a result, it doesn't increase energy efficiency. To overcome this weak point and reduce energy consumption, Hill, Culler [6] and El-Hoiydi [7] proposed that by sending out a very long preamble, the receivers only need to wake up periodically in order to sense the activity using a low-level carrier for preamble sensing. Notably, El-Hoiydi proposed Preamble Sampling [7] that applies to ALOHA, and WiseMAC [8] that sends out a short preamble beginning with channel sampling, because a long preamble is not necessary when the sender knows the sampling schedule of the receivers. WiseMAC dynamically changes the preamble size according to the network traffic state.

Another protocol specifically designed for sensor networks is S-MAC [2]. The basic idea of this single-frequency contention-based protocol is to divide the time into fairly large frames. Each frame has two parts: an active part and a sleeping part. During the sleeping part, a node turns off its radio to conserve energy. During the active part, it can communicate with its neighbors and send any messages queued during the sleeping part. S-MAC needs some synchronization, but this is not as critical as in TDMA-based protocols, because the time scale is much larger. The S-MAC protocol essentially provides a trade-off between the energy used for throughput and latency. The throughput is reduced because only the active part of the frame is used for communication. The latency is increased because a message-generating event may occur during the sleeping part. In this case, the message will be queued until the start of the next active part.

To solve this problem, the T-MAC protocol [3] was proposed. T-MAC, which adds flexibility to S-MAC's duty cycle, can be adapted to the network traffic state. In the T-MAC protocol, each node periodically wakes up to communicate with its neighbors, and then goes to sleep again until the next frame. Meanwhile, new messages are queued. A node will keep listening, and potentially transmitting, as long as it is in an active period. An active period ends when no activation event has occurred before a time-out. Therefore, T-MAC is more energy efficient than S-MAC. Like T-MAC, Dynamic S-MAC can also be adapted to the network traffic state. However, the flexibility of a Dynamic S-MAC extends to the entire frame, whereas that of the T-MAC is limited to the active interval.

3. DYNAMIC S-MAC DESIGN

3.1 Algorithm Description

The primary goal of a Dynamic S-MAC is energy efficiency. The main problem of energy efficiency is idle-listening. To reduce this idle-listening, S-MAC was designed to have a duty cycle, in order to make an energy efficient MAC protocol. However, it has a weak point, because it has a fixed duty cycle. Due to this fixed duty cycle, S-MAC cannot adapt to traffic variations.

In the case of the T-MAC, if a message is not generated before the time-out in the active period, this problem can be solved by changing to sleep mode.

Second, figure 2 shows that S-MAC has unnecessary active intervals. When the burst messages are generated in part A and there are no messages in parts B and C, unnecessary active intervals occur which decrease energy efficiency.

Dynamic S-MAC is a protocol that dynamically changes the entire frame size by adjusting the
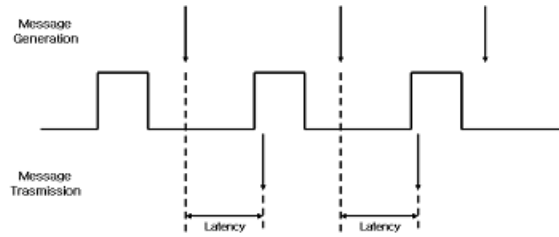
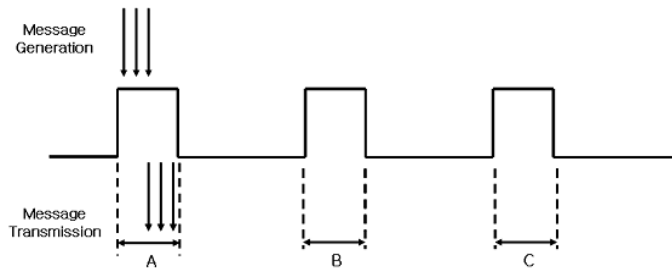Fig. 1. Latency increased by data generation

Fig. 2. Energy consumption by network traffic cohesion

network frame states.

The fundamental idea is to decide whether the interval is to be spent in the active or sleep state by means of a *flag* that records the data generation, transmission and reception. Figure 3 shows that the active interval is determined by the *flag* value. If the *flag* value is 0 in the prior active interval, the next active interval has a sleeping schedule. On the other hand, if the *flag* value is 1 in the prior active interval, the next active interval has an active schedule.

Moreover, the dynamic S-MAC changes the *flag* value to 1 in this sleeping active interval, so the next active interval can sense the messages. In other words, our protocol toggles the sleeping schedule in the active interval in cases in which no messages are generated.

As shown in figure 2, if burst traffic is generated in a special interval, the entire frame length is reduced, as shown in Frame C in Figure 3, for the purpose of reducing the message latency. In cases in which no message is generated, as in the case of Frames A, B and D, the entire frame length is prolonged, whereas in the case where a message is generated, as in the case of Frame C, the frame interval is reduced. In this manner, a dynamic S-MAC can dynamically control the entire frame length according to the network traffic state.
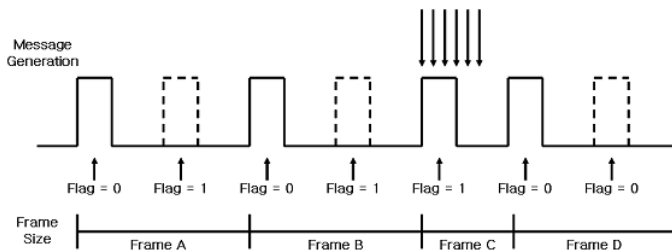

Fig. 3. Dynamic S-MAC operation and changing of the frame size

Regarding data transmission, the first packet is used to set the *flag* value of the neighbors to 1, so there is a slight delay. However, this reduces the average latency, because the reduced frame length is used for subsequent packets.

3.2 Synchronization

The Dynamic S-MAC uses the *flag* value to determine the active or sleep intervals. So, if the *flag* value is 0, the frame size is doubled. Therefore, when the node synchronizes with its neighbors, the dynamic S-MAC fundamentally operates like the S-MAC. Especially, when the SYNC packets are sent to the neighbors, they contain the schedule for cases in which the *flag* value is 1.

Figure 4 shows the process wherein node B receives a SYNC packet to be sent to node A and synchronizes with node A. Node B synchronizes with its neighbors according to the schedules in cases in which the *flag* value is 1.
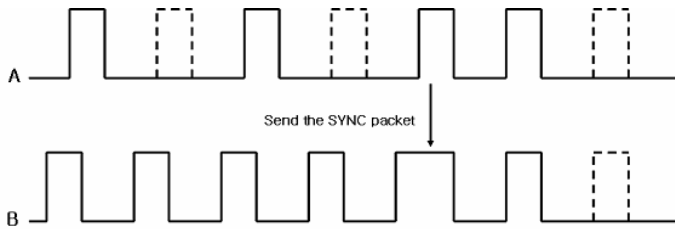

Fig. 4. Dynamic S-MAC synchronization

3.3 Operation with neighbors

Figure 5 shows the schedules of the neighbor nodes in instances of transmission, after synchronizing with the neighbors.

In State 1, node 3 sends an RTS packet to node 2 (requesting permission?) to transmit. If node 2 is ready to receive the messages, it sends a CTS packet to node 3. As in State 2, nodes 2 and 3 send and receive DATA/ACK packets to and from each other. At this time, the flag value remains continuously 1, so nodes 2 and 3 function using half of their usual frame size. In this situation, nodes 1 and 4 enter sleep mode, because of the *duration* in the header of nodes 2 and
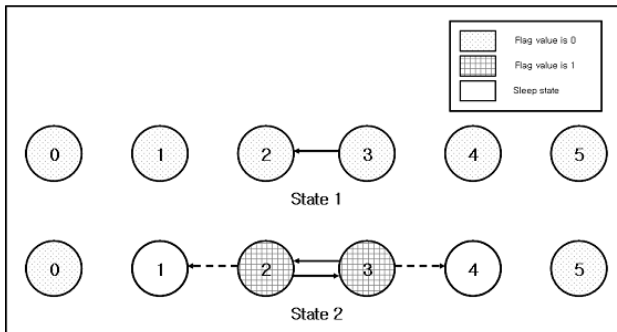

Fig. 5. Operation with neighbor nodes

3's DATA/ACK packets. As a result, nodes 0 and 5 don't send or receive messages and maintain an established frame length. Finally, in the message transmission, the node doesn't weigh with other nodes.

3.4 Protocol Analysis

Before experimenting with protocols S-MAC and dynamic S-MAC, we analyzed the performance of the dynamic S-MAC protocol, and compared it with that of the S-MAC protocol.

First, we describe the terminologies used in this paper.

- **Carrier sensing delay** : The latency incurred due to the carrier sensing procedure when a sensor contends for the channel. We denote the average value of the carrier sense delay by t_{cs}, which is determined by the contention window size.
- **Transmission delay** : The transmission time related to the channel bandwidth, packet length and coding scheme. Therefore, we assume a fixed packet length and denote the transmission delay by t_{tx}
- **Sleep delay** : The latency incurred due to the periodic sleeping of each sensor. The latency incurred by the periodic sleeping algorithm is denoted as ts in this context.
- **Frame length** : We denote the entire frame length by T_f.

In this section, we look at the latency for a packet transmission between two neighboring nodes that are one hop away from each other. Due to the channel capacity limitation, we ignore the propagation delay and the processing delay.

1) Latency of S-MAC: When a node sends to its neighboring nodes using the S-MAC protocol, it takes carrier sensing time and latency by the end of the sleep interval. Therefore, we can obtain the latency of a single-hop communication experienced by the network as follows:

$$D = t_{cs} + t_{tx} + t_s \tag{1}$$

The sleep delay ts is a random variable between $(0, Tf)$. The average latency for one-hop communication in the S-MAC is

$$E\,[D] = E[t_{cs}+t_{tx}+t_s] = t_{cs} + t_{tx} + T_f\,/\,2 \tag{2}$$

2) Latency of the Dynamic S-MAC: For the Dynamic S-MAC, as described above, we assume two duty cycle levels. Let $p1$ and $p2$ denote the probabilities that $Tf\ and\ Tf/2$ are the adjusted duty cycles in the algorithm, respectively. We have

$$E\,[D] = E\,[\,t_{cs} + t_{tx} + t_s\,] \tag{3}$$
$$= t_{cs} + t_{tx} + T_f(p_1/2 + p_2/4) \tag{4}$$
$$= t_{cs} + t_{tx} + T_f(2p_1 + p_2)/4 \tag{5}$$

In this time, $p1 + p2$ is 1.

$$2p_1 + p_2 < 2 \tag{6}$$
$$(2p_1 + p_2)/4 < 1/2 \tag{7}$$
$$E[D]_{Dynamic\ S\text{-}MAC} < E[D]_{S\text{-}MAC} \tag{8}$$

From the above analysis, it is easy to conclude that ts of the Dynamic S-MAC is smaller than that of the S-MAC. Thus, it can be stated that the average latency is decreased in the Dynamic S-MAC.

4. SIMULATION RESULTS

4.1 Simulation configuration

We used the NS2 (Network Simulator) [9] made by the LBNL (Lawrence Berkeley National Laboratory). NS2 is a very reliable simulator that is suitable for verifying the efficiency of wireless sensor networks.

In order to concentrate on the inherent properties of the S-MAC and dynamic S-MAC, we performed the tests on a simple two-hop network topology, as illustrated in Figure 6. The general simulation parameters are given in Table 1.
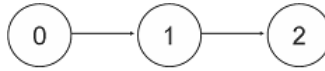

Fig. 6. Two-hop network topology

Table 1. General Simulation Parameters

Item	Value
Topology Size	1000m × 1000m
Initial Energy	1000 J
Transmit / Receive Power	0.5 W / 0.5 W
Active / Sleep Power	1.0 W / 0.0 W
Packet Size	20 Byte
Packet Interval	30 ms

4.2 Power Performance Simulation

Figure 7 is a graph that shows the energy consumption of the S-MAC and dynamic S-MAC in cases in which no messages are generated.

The Dynamic S-MAC has a larger frame length than the S-MAC, because the *flag* value is 0 since no data is generated. So, the sleep interval can be increased in the entire frame and, therefore, energy efficiency is increased.

Figures 8 and 9 compare the energy efficiency of the S-MAC and dynamic S-MAC in cases in which the packet intervals are 30ms and 5ms, respectively.

In Figure 8, the packet interval is relatively large, so it increases the intervals for which the *flag* value is 0. As a result, the dynamic S-MAC is more energy efficient than the S-MAC.

On the other hand, in Figure 9, it increases the intervals for which the *flag* value is 1, because

the packet interval is relatively small. In this case, the dynamic S-MAC has the same energy efficiency as the S-MAC, because they both have the same frame length.
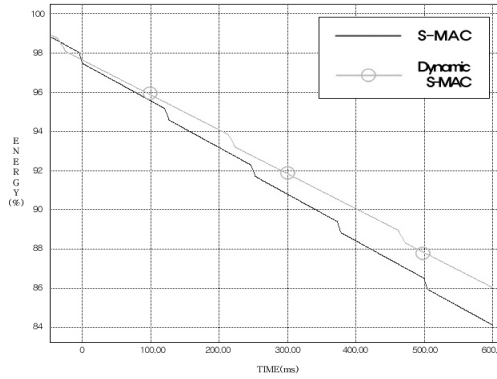

Fig. 7. The energy consumptions of S-MAC and dynamic S-MAC in a case where no messages are generated
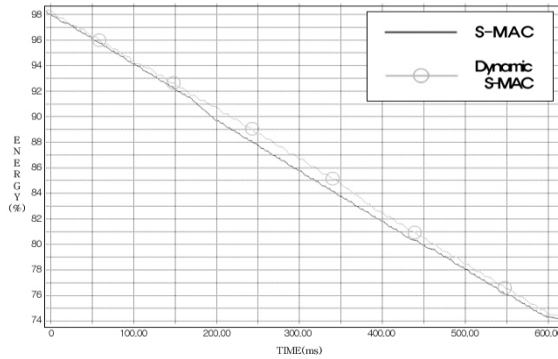

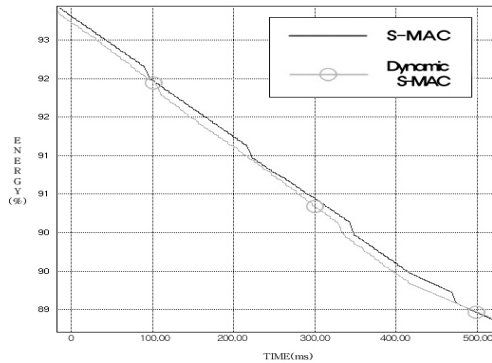Fig. 8. Energy efficiency of the S-MAC and dynamic S-MAC in a case where the Packet Interval is 30ms

Fig. 9. Energy efficiency of S-MAC and dynamic S-MAC in a case where the Packet Interval is 5ms

4.3 Latency Performance Simulation

In this simulation, we estimate the latency performance in a case in which the S-MAC and dynamic S-MAC use the same frame length.

Figure 10 shows the latency performance of S-MAC and dynamic S-MAC in the cases where they both use the same frame length and where the dynamic S-MAC uses twice the frame length of S-MAC.

When the dynamic S-MAC encounters an equal frame length as the S-MAC, then the greater the number of large packet intervals, the greater the latency of the dynamic S-MAC increases; even to near twice of that of the S-MAC. This is because the *flag* value frequently becomes 0, since packet generation is not incessant. Consequently, the dynamic S-MAC will have twice the frame size of the S-MAC. Therefore, the larger the packet interval, the more the latency is increased. However, we can verify that the latency decreases with decreasing packet intervals. Because this increases the probability that the *flag* value will become 1, the dynamic S-MAC will operate using the same frame size as the S-MAC. As a result, the dynamic S-MAC has the same latency performance as the S-MAC.

When the dynamic S-MAC encounters half the frame length of the S-MAC, then the larger the packet interval, the closer the latency of the dynamic S-MAC comes to that of the S-MAC. This also results from the fact that the *flag* value frequently becomes 0, because packet generation is not incessant. As a result, the dynamic S-MAC has the same frame size as the S-MAC. However, as the packet interval decreases, the latency performance of the dynamic S-MAC becomes increasingly superior to that of the S-MAC. Because this increases the probability of the *flag* value becoming 1, the dynamic S-MAC operates using half the frame size of the S-MAC. Therefore, it can be stated that the dynamic S-MAC has better latency performance than S-MAC.
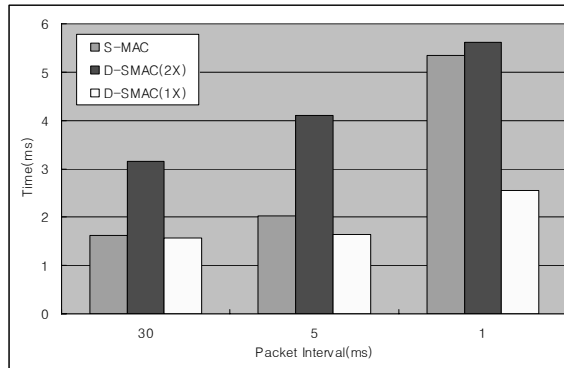

Fig. 10. Latency performance of S-MAC and dynamic S-MAC

5. CONCLUSION

The S-MAC solves the problem of idle-listening using the static duty cycle, but it is not efficient in cases in which various changes in network traffic occur. The Dynamic S-MAC adds flexibility to the S-MAC and can be more efficient with respect to network traffic. In the simulation, the dynamic S-MAC showed increased energy efficiency and latency performance, because it has a more flexible frame size than the S-MAC.

However, in the latency performance simulation, different results were obtained for the two types of dynamic S-MAC. We expect that this problem can be solved by finding the appropriate *flag* value.

REFERENCES

[1]   IEEE Standard 802.11, "Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) Specifications", 1999.

[2]   W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks", In 21st Conference of the IEEE Computer and Communications Societies (INFOCOM), Vol.3, June, 2002, pp.1567-1576.

[3]   T.van Dam And K.Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks", in Proc.ACM Sensys 2003, November, 2003, pp.171-180.

[4]   P. Havinga and G. Smit,"Energy-efficient TDMA medium access control protocol scheduling",In Asian International Mobile Computing Conference (AMOC 2000), November, 2000, pp.1-9.

[5]   S. Singh and C. Raghavendra, "PAMAS: Power aware multi-access protocol with signalling for ad hoc networks", ACM SIGCOMM Computer.

[6]   J. Hill and D.Culler.Mica, "A wireless platform for deeply embedded networks", IEEE Micro, Vol.22, November, 2002, pp.12-24.

[7]   A.El-Hoiydi, "Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks", in Proc. IEEE ICC 2002, Vol.5, April, 2002, pp.3418-3423.

[8]   A.El-Hoiydi, J.-D. Decotignie, C.Enz and E.Le Roux, "Poster abstract : WiseMAC, an Ultra Low Power MAC Protocol for the WiseNET Wireless Sensor Network", in Proc.ACM Sensys 003, November, 2003.

[9]   The CMU Monarch Project, "The CMU Monarch Project's Wireless and Mobility Extensions to NS.".

Dae-Suk Yoo

He received a BS degree in Computer Engineering from Univ. of Incheon in 2008. From 2008, he has been working for Samsung SDS. His research interests include RFID/Sensor Networks, Wireless MAC.

Seung Sik Choi

He received a B.S. degree in Electronics engineering from Yonsei University in 1988, and M.S. and Ph.D degrees in Electrical and Electronics Engineering from the Korea Advanced Institute of Science and Technology (KAIST) in 1990 and 2002, respectively. From 1990 to 2003, he was at the Communication Network Lab. in Korea Telecom, where he worked on B-ISDN/ATM switch and wireless LAN systems. From 2004, he has been a professor in the department of computer engineering, University of Incheon. His interests include communication networks, sensor networks, resource management in CDMA systems, protocol design and medium access control.