A Pseudo-Random Asynchronous Duty Cycle MAC Protocol in Wireless Sensor Networks

Hanjin Lee, Jaeyoung Hong, Suho Yang, Ingook Jang, and Hyunsoo Yoon

***Abstract*—We propose a pseudo-random asynchronous duty cycle MAC protocol in wireless sensor networks. The proposed protocol adopts a hash function to determine the next wake-up times. As the next wake-up times are known in advance, the sensor nodes do not need to remain active until their intended receivers wake up. Given an end-to-end delay requirement, the proposed protocol significantly decreases energy consumption by reducing the idle listening time.**

***Index Terms*—Duty cycle, asynchronous, hash function, idle listening.**

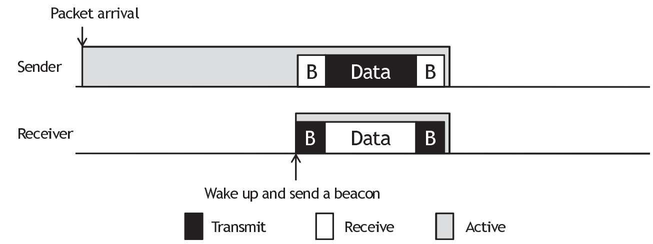

Fig. 1. Operation of RI-MAC.

I. Introduction

In wireless sensor networks (WSN), energy consumption is one of the most important factors because it is difficult to recharge or replace the battery of each sensor node. Therefore, most medium access control (MAC) protocols in WSNs employ the duty cycling technique, in which sensor nodes turn their radio on and off repeatedly, to save energy.

There are two types of duty cycle MAC protocols: synchronous and asynchronous. In synchronous duty cycle MAC protocols such as S-MAC [1] and T-MAC [2], sensor nodes wake up and sleep at the same point in time. However, these protocols require time synchronization, which causes control message overhead and makes sensor nodes complex and expensive. On the other hand, in asynchronous duty cycle MAC protocols, each sensor node wakes up and sleeps independently. Thus, time synchronization is not necessary. Most asynchronous duty cycle MAC protocols adopt a random wake-up interval in order to avoid repeated collisions. When several sensor nodes wake up at the same time, collisions may occur due to simultaneous transmissions. If these sensor nodes adopt the same wake-up interval, collisions may occur repeatedly whenever they wake up and try to send data. Given that sensor nodes wake up at different times with random wake-up intervals, it is necessary to ensure that a sender and its intended receiver are active at the same point in time to transmit data. To do this, preamble-based protocols were proposed in [4][5][6]. However, preamble transmission occupies the wireless medium for a long time, which decreases throughput and increases delay. Recently, a receiver-initiated duty cycle MAC protocol (RI-MAC) was proposed in [7]. In this protocol, a sender wakes up and remains active until the intended receiver sends a base beacon. The receiver sends a base beacon whenever it wakes up. After the sender receives the beacon, data transmission is started. Though this protocol improves preamble-based protocols in terms of energy consumption and delay, the idle listening time is considerable nonetheless.

In this letter, a pseudo-random asynchronous duty cycle MAC protocol to reduce the energy consumption caused by idle listening is proposed. It adopts a hash function to determine the next wake-up times that are not periodic in an effort to avoid repeated collisions. The rest of this letter is organized as follows: Section II discusses related works. In Section III, the pseudo-random asynchronous duty cycle MAC protocol is proposed. Its performance is compared with that of RI-MAC in Section IV. Finally, we conclude the letter in Section V.

II. Related Works

In STEM [3], there are separate radios for data and wake-up signals in sensor nodes. A sensor node turns on the wake-up radio in order to wake its intended receiver. This protocol requires two radio modules, which makes sensor nodes expensive. In B-MAC [4], when a packet arrives at a sender, it wakes up and sends a long preamble which lasts for longer than the sleep interval of its intended receiver. If the sender receives a response from the intended receiver after it finishes sending the preamble, it starts to send data. However, long preambles occupy the wireless medium for a long time, leading to high energy consumption and delay. In X-MAC [5], instead of a long preamble, a sender sends a sequence of short preambles. When one short preamble is received, the intended receiver responds immediately. Thus, the sender stops sending the preambles and starts to send data, which yields higher channel utilization than B-MAC. However, though it reduces the preamble transmission, the problem outlined in B-MAC remains. Wise-MAC [6] is also based on preamble transmission. However, after data transmission, a receiver sends an ACK message including the time remaining until its next wake-up time. Thus, the sender can wake up and start to send a preamble just before the receiver

Manuscript received September 15, 2009. The associate editor coordinating the review of this letter and approving it for publication was C.-F. Chiasserini.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (No. 2009-008364).

The authors are with the Department of Computer Science, KAIST, Republic of Korea (e-mail: hjlee@nslab.kaist.ac.kr).

Digital Object Identifier 10.1109/LCOMM.2010.02.091849

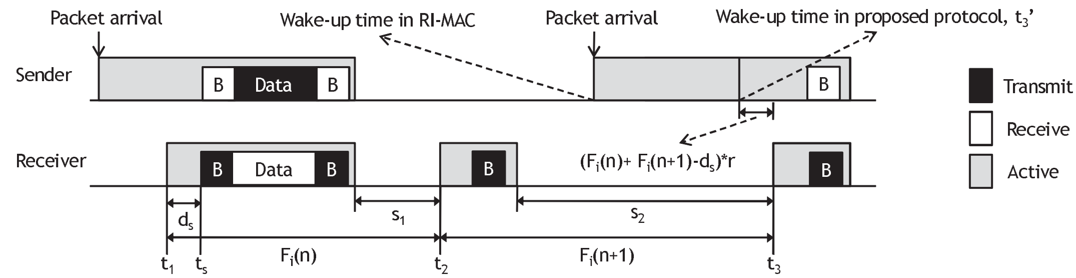

Fig. 2. Operation of the proposed protocol.

wakes up. As the time duration of the preamble transmission is short, Wise-MAC reduces energy consumption and improves channel utilization. However, repeated collisions can occur because this protocol assumes a periodic wake-up time. In addition, simultaneous preamble transmissions from hidden nodes degrade its performance.

In RI-MAC [7], as shown in Fig. 1, when a packet arrives at a sender, it wakes up and simply waits for a base beacon from its intended receiver. When the receiver wakes up, it sends a base beacon as an invitation for data transmission. If the sender receives the base beacon, data transmission is started. When the data is successfully received, the receiver sends a beacon as an acknowledgement (also used as an invitation for new data transmission). If a collision occurs, the receiver sends a beacon which includes the backoff window size (W). Consequently, senders perform a random backoff based on W before retransmitting data.[1] This protocol uses a short beacon message instead of preambles which waste the wireless medium. Thus, it decreases energy consumption and delay significantly. However, the idle listening time of senders is long. In other words, a sender should wake up and remain active until the intended receiver sends a beacon message.

III. Proposed Protocol

A. Determination of wake-up interval

We propose an efficient method for determining the wake-up interval. Sensor node i calculates the wake-up interval as follows:

$$F_i(n) = mod(Hash(n \oplus ID_i), T_{range}) + T_{mean} - T_{range}/2 \tag{1}$$

where n is a sequence number which is increased by 1 whenever sensor node i wakes up, ID_i is the identification of sensor node i, and $\oplus$ is the exclusive OR operation. $Hash(n \oplus ID_i)$ is the output of a hash function from the input of $n \oplus ID_i$. To avoid the repeated collisions mentioned in Section I, the hash functions in the proposed protocol must have the property of uniformity to randomize outputs. In addition, $mod(a, b)$ is the remainder of the division of a by b. It is used to limit the value of $F_i(n)$. T_{mean} represents the mean of $F_i(n)$, and T_{range} denotes the range of the wake-up interval.[2] Thus, each sensor node generates a pseudo-random wake-up interval, $F_i(n)$, ranging from $T_{mean} - T_{range}/2$ to $T_{mean} + T_{range}/2$.

[1]The *'base beacon'* refers to the beacon which the receiver sends only when it wakes up. We use the term *'beacon'* for the beacon which is used as an acknowledgement or for one that includes the backoff window size.

[2]T_{range} = the maximum value of $F_i(n)$ − the minimum value of $F_i(n)$

B. Operation of the Proposed Protocol

This section describes the operation of the proposed protocol based on RI-MAC.[3] As shown in Fig. 2, when a packet arrives at a sender, it wakes up and remains active until the intended receiver i sends a base beacon. The receiver wakes up at t_1 and sends a base beacon at t_s after the channel is clear. If a sender receives the base beacon, it starts sending data immediately. Unlike RI-MAC, in the proposed protocol, the base beacon includes a sequence number n and the time difference between the wake-up time (t_1) and the start time of base beacon transmission (t_s), denoted by d_s. Thus, the sender can calculate the last wake-up time of the receiver, $t_1 = t_s - d_s$. Both the sender and receiver calculate the next wake-up times of the receiver i, $t_2 = t_1 + F_i(n)$, $t_3 = t_1 + F_i(n) + F_i(n+1)$ continuing as far as is necessary. In other words, the sender recognizes all of the wake-up times of the receiver in the future. Thus, when a packet arrives later, the sender does not wake up immediately. It has the wake-up schedule of the receiver. Hence, it wakes up just before the receiver wakes up in order to save energy. This approach significantly reduces the idle listening time with very low additional overhead.

C. Handling Clock Drift

A clock in each sensor node runs at a different speed in what is referred to as clock drift. Therefore, $t_k (k > 1)$ as calculated by the sender and its receiver can be different in actuality. In general, the upper bound of clock drift is given in datasheets. For example, the upper bound of clock drift in Berkeley motes is 40 ppm (a clock in a sensor node can lose up to 40 μs per second) [8]. Therefore, in the proposed scheme, senders must wake up slightly earlier than the calculated wake-up time of receiver i. Receiver i wakes up at every $t_k (k > 1)$, while the senders wake up at $t'_k (k > 1)$ if they have data to send, as follows:

$$t'_k = t_s + (1 - r) \cdot \Big\{ \sum_{m=0}^{k-2} F_i(n + m) - d_s \Big\} \tag{2}$$

where r is the upper bound of clock drift. For example, in Fig. 2, the receiver wakes up at t_2 and t_3 while the sender wakes up at $t'_2 = t_s + \{F_i(n) - d_s\} - \{F_i(n) - d_s\} * r$ and $t'_3 = t_s + \{F_i(n) + F_i(n+1) - d_s\} - \{F_i(n) + F_i(n+1) - d_s\} * r$ if it has data to send.

[3]The proposed protocol can be also easily applied to other asynchronous MAC protocols such as B-MAC and X-MAC.

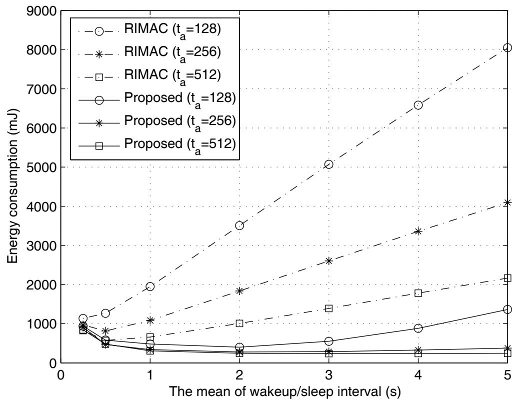

Fig. 3. Energy consumption according to the wake-up/sleep interval.

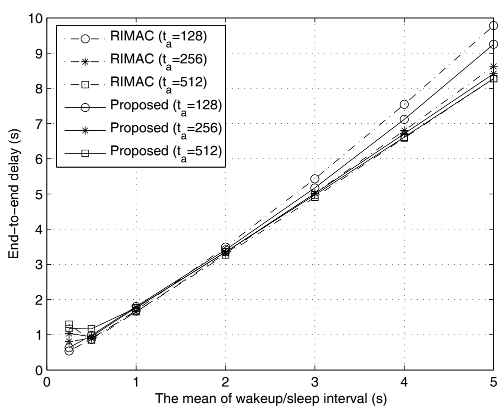

Fig. 4. End-to-end delay according to the wake-up/sleep interval.

IV. Performance Evaluation

To show the effectiveness of the proposed protocol, a simulation was carried out using the NS-2 simulator. Receiving, listening, transmitting and sleep modes consume 13.5 mW, 13.5 mW, 24.75 mW, and 15 μW, respectively [9]. We deploy 50 sensor nodes randomly in 1000 m $\times$ 1000 m area, and one sensor node is randomly selected as a sink. The transmission range is set to 250 m and channel model is a two-ray ground model. All sensor nodes except the sink independently generate packets and send them to the sink through multi-hop transmissions. Each node generates packets according to a Poisson process, i.e., the packet inter-arrival time is exponentially distributed. The data packet size is set to 128 bytes[4], and the size of the base beacons is 6 bytes in RI-MAC; this size is increased by 2 bytes due to the inclusion of n and d_s in the proposed protocol. The simulation time is 10000 s and the results of 30 independent scenarios are averaged for each wake-up interval. T_{mean} is set to 0.25, 0.5, 1, 2, 3, 4, and 5 s, and T_{range} is set to $T_{mean}/2$. The value of r in (2) is set to 100 ppm.

Figures 3 and 4 show the energy consumption and the end-to-end delay, respectively. In these figures, t_a represents the mean of the packet inter-arrival time. The results of the proposed protocol are shown according to the mean of the wake-up interval; those of RI-MAC are shown according to the mean of the sleep interval.[5] Here, it is important to compare the minimum energy consumption when an end-to-end delay requirement is given, rather than comparing two protocols at a wake-up/sleep interval. For example, assuming that the mean of the packet inter-arrival time is 256 s and the end-to-end delay requirement is 3.5 s, as shown in Fig. 4, to satisfy this delay requirement, the mean of the wake-up/sleep interval should be less than or equal to 2 s. In this case, RI-MAC has the minimum energy consumption, approximately 900 mJ, when the sleep interval is 0.5 s. On the other hand, the minimum energy consumption of the proposed protocol is approximately 285 mJ when the wake-up interval is 2 s. The proposed protocol consumes less energy than RI-MAC due to the significant reduction of the idle listening time. When the wake-up/sleep interval is small, energy consumption and the end-to-end delay increase slightly because sensor nodes wake up frequently and many beacon messages are transmitted. In addition, it is clear that energy consumption and end-to-end delay increase as the mean of the inter-arrival time decreases.

V. Conclusion

A pseudo-random asynchronous duty cycle MAC protocol for WSNs is proposed. It adopts a hash function to determine the next wake-up times considering clock drift. A sender and its intended receiver can share their wake-up times with very low additional overhead. Thus, the idle listening time is significantly reduced. Simulation results show that the proposed protocol significantly reduces energy consumption when an end-to-end delay requirement is imposed.

References

[1] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Networking*, vol. 12, no. 3, pp. 493-506, June 2004.

[2] T. Dam and K. Langendoen, "An adaptive energy efficient MAC protocol for wireless sensor networks," in *Proc. ACM Sensys 2003*, pp. 171–180, Nov. 2003.

[3] C. Schurgers, V. Tsiatsis, and M. Srivastava, "STEM: topology management for energy efficient sensor networks," in *Proc. IEEE Aerospace Conf.*, 2002.

[4] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. ACM Sensys 2004*, pp. 95–107, Nov. 2004.

[5] M. Buetter, G. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proc. ACM Sensys 2006*, pp. 307-320, 2006.

[6] A. El-Hoiydi and J. Decotignie, "WiseMAC: an ultra low power MAC protocol for multi-hop wireless sensor networks," in *Proc. ALGOSENSORS 2004*, pp. 18–31, July 2004.

[7] Y. Sun, O. Gurewitz, and D. Johnson, "RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," in *Proc. ACM Sensys 2008*, pp. 1–14, 2008.

[8] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timing-sync protocol for sensor networks," in *Proc. ACM Sensys 2003*, pp. 138–149, 2003.

[9] ASH Transceiver TR1000 Data Sheet, RF Monolithics Inc., Dallas, TX. [Online]. Available: http://www.rfm.com/

[4]The data packet size has little impact on the performance. Hence, the performance results according to the size are omitted. This is why the total time duration of packet transmission is much shorter than the simulation time.

[5]The wake-up interval represents the time difference between two consecutive wake-up time points, e.g., $F_i(n)$ and $F_i(n+1)$ in Fig. 2. On the other hand, the sleep interval used in RI-MAC is the time duration for which a sensor node is in sleep mode, as denoted by s_1 and s_2 in Fig. 2.