An Adaptive MAC (A-MAC) Protocol Guaranteeing Network Lifetime for Wireless Sensor Networks

Yongsub Nam, Hojin Lee, Hakyung Jung, Taekyoung Kwon and Yanghee Choi

School of Computer Science and Engineering
Seoul National University
Email: {ysnam, lumiere, hkjung, tk, yhchoi}@mmlab.snu.ac.kr

***Abstract*— Wireless sensor networks (WSNs) have a wide range of potential applications, a majority of which may be required to survive for a pre-configured lifetime since it is hard or costly to re-deploy sensor nodes to where a *sensing hole* has taken place. A sensing hole may occur when specific nodes suffer more traffic load than others and thus exhaust initially supplied energy. We propose an adaptive MAC (A-MAC) protocol for WSNs to tackle the sensing hole problem, which keeps the entire network operating for a required lifetime. The main concern in designing A-MAC is two-folded: guaranteeing the pre-configured network lifetime, and reducing end-to-end latency. In order to achieve both goals, A-MAC introduces an adaptive duty cycle depending on ratio of the remaining energy to the initially supplied energy considering the pre-configured lifetime. The more energy a node has, the more frequently the node will wake up and hence fasten relaying data. As a consequence, each node is expected to run out of energy around the end of the pre-configured network lifetime. Also, nodes with more energy are utilized to reduce the end-to-end delay. Simulation results exhibit significantly lower latency of A-MAC while guaranteeing the pre-configured network lifetime.**

***Index Terms*— Wireless sensor network; medium access control (MAC); network lifetime; duty cycle adaptation; cross-layer;**

I. INTRODUCTION

A wireless sensor network (WSN) comprises a set of sensor nodes deployed in an area of interest. Sensor nodes are normally deployed in an ad hoc manner, and collecting data, processing, or relaying it. WSNs have wide-ranging applications, including environmental monitoring, medical, and target tracking systems. In this paper, we focus on WSNs whose environments are too harsh to recharge battery or replace energy-depleted nodes. These applications may have following requirements, which should be considered in designing a MAC protocol.

- **Energy conservation**: Energy is the most crucial but scarce resource in WSNs. Therefore, it is important to utilize the given energy efficiently. Currently, this goal is readily achievable by adopting a periodic listen/sleep schedule.
- **Pre-configured network lifetime**: Although most of previous work does not focus on this issue, the pre-configured lifetime is also significant. In many cases, people should re-deploy sensor nodes when they do not receive sensory data from the dead nodes, so-called sensing hole. However, it is not an economical method because of the harsh environment of the target region. The lifetime of the network should be predictable and configurable, so that the operators can determine when to deploy new sensor nodes.
- **End-to-end latency**: Target tracking or monitoring events are likely to be time-bounded, which means the events should be transmitted to the sink node as quickly as possible. Therefore, reducing the end-to-end delay is important in these applications.
- **Quality of Surveillance (QoSv)**: QoSv is introduced in [10] and represents how much area is covered with sensor nodes. If network partitioning occurs or a sensing hole takes place since some nodes run out of energy, the network cannot guarantee QoSv. In order to ensure QoSv, it is significant for all nodes to survive for a required network lifetime.

This paper presents an adaptive MAC (A-MAC) protocol designed for WSNs. To the best of our knowledge, this is the first approach to guarantee the pre-configured network lifetime. While ensuring the pre-configured network lifetime is the primary goal in our design, we also focus on reducing the end-to-end latency. To obtain these goals, each node sleeps and wakes up periodically, and also changes its sleep/wakeup cycle adaptively based on the remaining energy of each node. Accordingly, each node dies almost simultaneously, achieving better QoSv (or sensing coverage).

In this paper, we first introduce the related work, and in Sections 3 and 4, we detail our proposed scheme and a cross-layer approach encompassing routing, respectively. The numerical results are shown in Section 5 and concluding remarks are given in Section 6.

II. RELATED WORK

A number of research efforts have focused on the MAC protocol in WSNs. Since energy-efficiency has been the main concern in designing MAC protocols in sensor networks, most of them adopt periodic sleep/wakeup cycle for energy conservation. S-MAC [1] is a representative MAC protocol based on a periodic sleep/wakeup model, which reduces energy consumption due to idle listening. However, since S-MAC trades throughput and latency for energy efficiency, it will incur large delay especially in cases of the target tracking applications. In addition, S-MAC does not take the network lifetime into consideration.

Approaches to reduce latency of S-MAC have been suggested in [2] and [3]. These protocols are based on periodic sleep/wakeup cycles and change the duty cycle of each node depending on the data traffic. They show lower delay than S-MAC while maintaining similar or a little bit less energy-efficiency. D-MAC [4] also focuses on reducing end-to-end latency. It adopts the data gathering tree and staggered scheduling, so that intermediate nodes wake up in due order of data forwarding to the sink node, resulting in the less delay. However, these protocols only increase wake-up time portion of the duty cycle for less latency, so that they sacrifice energy-efficiency, resulting in even shorter network lifetime than S-MAC. Thus, these protocols are not suitable for the habitat monitoring or target tracking applications.

ASAP [8] is another contention-based MAC protocol, exploiting to prolong the network lifetime while satisfying the latency requirements. ASAP uses a hash function to decide its listen/sleep schedule, i.e., hashing the node identifier (e.g., MAC address), to determine its listen slot in a frame. A node which has data to send to its neighbor node wakes up at the neighbor's listen slot and transmits it. The listen/sleep schedule changes in run-time depending on the node density and its residual energy. Because ASAP employs staggered scheduling, it can notably reduce the end-to-end delay in the multi-hop case. However, broadcasting data requires repeated transmissions at the listen slot of each neighbor node, which causes much more control traffic. Also, in terms of the network lifetime, ASAP does not guarantee the pre-configured network lifetime but only tries to prolong the network lifetime in a best-effort manner.

B-MAC [7] is also designed to minimize idle listening energy consumption. B-MAC contains a small core of media access functionality, such as clear channel assessment (CCA) and packet backoffs for channel arbitration, link layer acknowledgements for reliability, and low power listening (LPL) for low power communication. In B-MAC, each node wakes up periodically to check for channel activity. If activity is detected, the node powers up and stays awake for the time required to receive an incoming packet. To ensure that all packets are heard by the nodes, packets are sent with a preamble whose transmission/reception time is longer than the check interval. For a given network configuration (each node's neighbor node size and the ideal sampling rate), B-MAC's parameters such as check interval can be calculated and hence its lifetime can be estimated. B-MAC assumes identical operation (i.e., sampling rate) of all the sensor nodes. Therefore its lifetime analysis can be applied to only special applications of WSNs.

On the other hand, the polling-based MAC protocols, [5] and [6] are suitable for deterministic data gathering applications. These protocols pursue reducing energy consumption by maximizing sleep time of each sensor node. However, they also do not consider remaining energy of each node or the network lifetime. Furthermore, since polling-based MAC protocols are vulnerable to the channel error and require elaborate time synchronization among nodes, they tend to show worse scalability than contention-based protocols.

III. A-MAC DESIGN

We primarily focus on ensuring the configured lifetime determined a priori. In addition, since the end-to-end delay is also crucial in our target applications, we also seek to reduce end-to-end latency. If every node shares the traffic load fairly and dies at the end of the pre-configured lifetime simultaneously, our goals can be achieved. Thus, we adopt a strategy where each node consumes energy approximately equally. Since we aim at reducing end-to-end latency while maintaining the pre-configured lifetime, we employ a duty cycle adaptation mechanism. Our proposed protocol is based on S-MAC's periodic listen/sleep schedule, and each sensor node changes its duty cycle depending on its energy consumption rate, while in S-MAC, the duty cycle is fixed in each node once it is configured. Thus, in A-MAC, the node with relatively higher remaining energy wakes up more frequently and serves more for the network. This way, data traffic load over the network lifetime will be distributed almost equally to each node, resulting in the fairness of each node's energy consumption rate. This also leads to better sensing coverage or QoSv, which mainly relies on the number of remaining active sensor nodes. In designing A-MAC protocol, we assume the network is densely deployed and the sensing events occur in a low frequency.

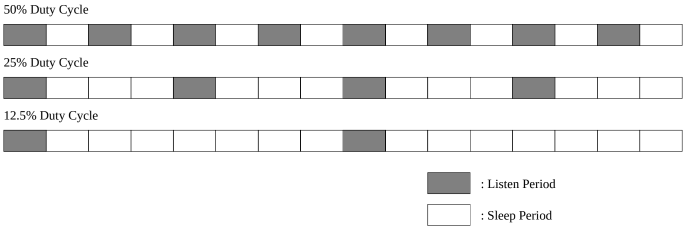


Fig. 1. Example of the duty cycle distribution

A. Protocol Overview

A-MAC employs a periodic listen/sleep mechanism. An example of the duty cycle distribution of three nodes is described in Fig. 1. One cycle of a sleep and a listen period is referred to as a *Superframe*. As shown in Fig. 2, a superframe comprises a listen and a sleep period, and the listen period is composed of SYNC and RTS/CTS time slots. The length of the listen period is fixed in A-MAC, so that the duty cycle only depends on the length of the sleep period. During the listen period, the SYNC information and RTS/CTS packets are exchanged. When the RTS/CTS message is successfully exchanged, both the sender and the receiver should wake up at the sleep period and send/receive data.

A node initially determines its own listen/sleep schedule and periodically broadcasts it in the *SYNC* message. The other nodes listen for this synchronization information. If

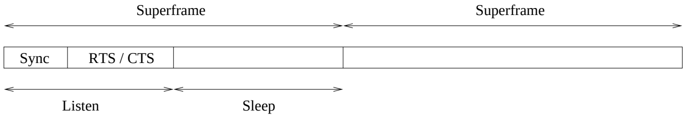


Fig. 2. A-MAC Superframe structure

the node hears a schedule from another node, it adopts the received schedule as its own, like in S-MAC. Note that each node follows the adopted schedule only at the beginning. This self-organization process will be finished in the initial synchronization period.

After the self-organization phase, the network operation phase starts. In this phase, a node changes the duty cycle depending on the remaining energy. The detailed mechanism of the duty cycle adaptation is given in the next section. The SYNC message consists of three main fields: the source address, the next wake-up time and the listen/sleep schedule. The source address is the address of the node sending the SYNC frame. The next wake-up time and the sleep schedule fields are announced to inform when the node will be active again and how often the active period will be, respectively.

This way, each node keeps track of all of the one-hop neighbors' schedules. Each node wakes up during the neighbor's schedule if packets should be transmitted to that node. In S-MAC, sensor nodes with the same schedule form a *virtual cluster* and the nodes in the border follow both clusters' schedules. On the other hand, in A-MAC, nodes hardly form virtual clusters because each node dynamically changes its own schedule depending on its energy consumption rate. As a consequence, the schedules of one-hop neighbors should be maintained.

Fig. 3 illustrates a data transmission from the source node (A) to the sink (E). The percent next to the node id is the duty cycle; for example, 25% node will have the sleep period three times as long as the listen period. Each node basically follows its own listen/sleep schedule, and if a node has a packet to send, it wakes up on the next hop's listen period, and after RTS/CTS exchange, it transmits data to the next-hop node. In this manner, data is relayed to the sink. On the other hand, if the node does not have packets to send, it does not wake up on the neighbor's listen period.

In A-MAC, too frequent broadcast of the SYNC message can lower the network performance. If two or more nodes send the SYNC messages simultaneously, they may collide with each other. In such a case, the changes in the duty cycle cannot be informed to neighbors. To alleviate this problem, each node normally broadcasts the SYNC message once in every n superframes of the minimum duty cycle neighbor; we choose ten for n in our experiments. Also, a node randomly chooses when to send the SYNC message among n superframes. By doing so, the SYNC message storm problem can be mitigated.

In addition, when a node wants to adjust its duty cycle, it can only double it or cut it in half. This is to make the sensor nodes more tolerant of the SYNC message loss. Also with the help of superframe synchronization which will be given in the next section, any two nodes are active simultaneously in the beginning of the superframe of the lower duty cycle node. Thus, when a node does not hear its neighbor's new schedule, its data transmission fails only when the neighbor decreases its duty cycle; however, the next try will succeed.

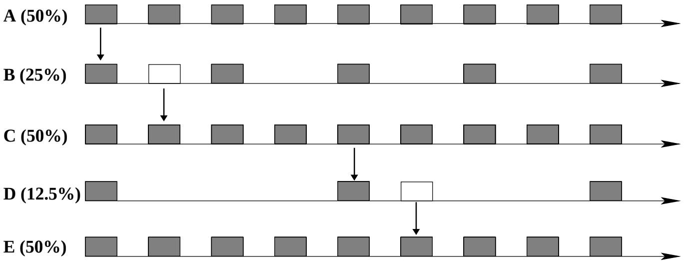


Fig. 3. Example of data transmission in A-MAC

B. Duty Cycle Adaptation

In terms of the duty cycle control, we mainly consider the remaining energy level of each node. Basically, the node with less energy should sleep more in order to balance the energy level of each node. To calculate the duty cycle precisely, we take into account both the remaining energy level and the remaining lifetime.

Let T_{conf} and T_{elap} denote the pre-configured network lifetime and elapsed time, respectively. Then, the ratio of T_{elap} to T_{conf} represents the ideal energy consumption rate, because if a node consumes the energy in that rate the node will die exactly at the end of the required lifetime. However, the traffic load is different among nodes and varies over time. So, fixing the same energy consumption rate for each node is not possible. Therefore, we choose to adjust the duty cycles of sensor nodes dynamically, which can make a sensor node consume its energy approximately at the ideal consumption rate. Let δ in Eq. (1) denote the difference between the ideal and the currently calculated energy consumption rate.

$$\delta = \frac{T_{elap}}{T_{conf}} - \frac{E_{init} - E_{rem}}{E_{init}} \tag{1}$$

Here, E_{init} and E_{rem} stand for the initial energy and the remaining energy of a node, respectively. In other words, the second term of the right-hand side in Eq. (1) indicates the energy consumption rate of the node so far. If δ becomes larger than the upper threshold, which implies the node has more remaining energy than it needs to last for the required lifetime, the node doubles its duty cycle. On the other hand, if δ is smaller than the lower threshold, the node cannot survive until the pre-configured lifetime; thus, the node decreases its duty cycle by half. If δ is in-between, the current duty cycle is maintained. The reason for exponentially increasing/decreasing adaptation is to ease the superframe synchronization among nodes. In this manner, the energy consumption would approach to the ideal energy consumption rate. Another significant advantage of A-MAC is that it does not require to configure the initial duty cycle carefully. Here, we assume that there is a minimum duty cycle and it can make the node survive for a pre-configured network lifetime, no matter how much the traffic load is.

C. Synchronization

In addition to the fields in the original SYNC message, we need two more fields: the timestamp of the transmission time and the next starting time of the minimum duty cycle schedule. The timestamp of SYNC transmission time is used to correct clock skews among the nodes. The next starting time of the minimum duty cycle schedule is used to synchronize the starting time of each node's superframe. When a node changes its duty cycle, it starts a new superframe in the beginning of the next superframe of the minimum duty cycle. Without superframe synchronization, two nodes may not communicate with each other because if the starting times of the two schedules are different, they cannot receive the SYNC messages from each other. Note that a node's schedule is the same as that of the synchronizer only in the self-organization phase; it will change depending on the energy consumption rate in the network-operation phase. On the other hand, if the synchronizers of two nodes are equal, they will awake simultaneously at least once during the period of the minimum duty cycle.

IV. CROSS-LAYER APPROACH

Since A-MAC adapts its duty cycle independently of other nodes, we suggest a routing protocol running on top of A-MAC choose a path with more remaining energy rather than a path with the lowest hop count. Existing routing protocols for WSNs, e.g., [12], [13], and [14], consider energy consumption when they select the routing path. We believe these routing protocols will work well with A-MAC. However, they do not take into account balanced energy distribution. In this case, the potential of A-MAC may not be highlighted. Hence, we present two energy-aware routing algorithms, which can take advantage of A-MAC. These routing algorithms will reflect the changes in the duty cycles, finding out a new path with more remaining energy. The two possible metrics for routing are listed below. To compare the performance of these metrics with the traditional one, i.e., the minimum hop count, we use AODV [**?**] as the base routing protocol. We modify AODV to implement these routing algorithms. Although AODV exhibits a large overhead, it is expected to clearly show the difference among the routing metrics since it is based on a simple design.

- **Max-Min algorithm**: this routing algorithm seeks to find out the path whose minimum duty cycle of the nodes along the path is maximum. The duty cycle of the intermediate node with the minimum duty cycle is recorded in the RREQ/RREP messages while discovering the routes. In choosing the routing path, the path with the largest minimum duty cycle value is selected. If two routes have the same minimum duty cycle values, the route which is found with less delay is selected. This algorithm can distribute the load in diverse paths, considering the energy level of relay nodes. However, it can take a long detour when there exists a node with the high duty cycle but far away from the normal route, which deteriorates the delay performance as well as consumes far more energy.
- **Max-Avg algorithm**: the route with the largest average duty cycle is selected. The number of hops and the accumulated duty cycle values are stored in the RREQ/RREP messages and the average duty cycle is calculated at each hop. The path with the largest average duty cycle value is selected. Again, the delay is used for tie-breaking. This metric is an eclectic approach because it divides total duty cycle values of the nodes on the path by the number of hops. Therefore, this algorithm can mitigate the problem of the above Max-Min algorithm.

TABLE I

RADIO AND MAC PARAMETERS

Radio transmission range	30 m
Radio interference range	60 m
Radio bandwidth	20 Kbps
Duration of listen interval	115 ms
Difference upperbound	0.1
Difference lowerbound	0
Configured network lifetime	4000 $TimeUnit$

TABLE II

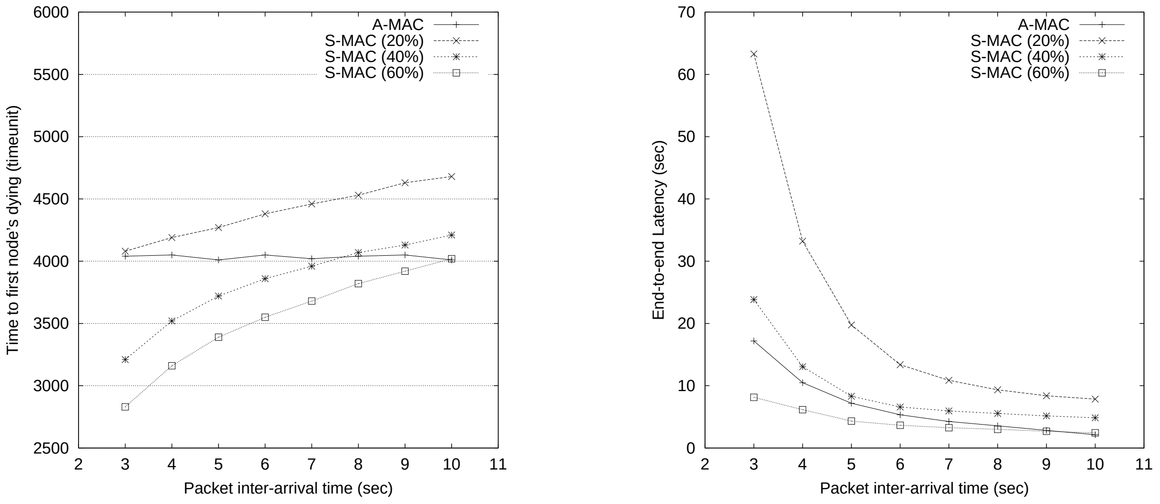
ENERGY PARAMETERS

Transmit power	0.660 Watts
Receive power	0.395 Watts
Idle power	0.350 Watts
Sleep power	0.001 Watts
Initial Energy	300 Joules

In our approach, when to trigger a new route discovery is an important issue, since the current route is still alive. Because the route discovery process usually floods the control messages, a periodic route discovery may cause an unnecessary control overhead. Thus, we introduce a sink-triggered route discovery. That is, the sink requests a new route discovery to the sender when it detects the average duty cycle value of the received packet goes below the average duty cycle at the previous route discovery by a certain threshold. This triggering is designed to work with the Max-Avg algorithm; the same approach can be applied to the Max-Min algorithm. The duty cycle value of each node on the path is accumulated and recorded in the packet header along with the number of hops. The sink keeps track of this duty cycle value, and when it falls below the average value at the previous route discovery by a certain threshold, the sink sends a route discovery trigger message to the sender. In this way, the routing path can change dynamically and the traffic can be distributed among nodes.

V. PERFORMANCE EVALUATION

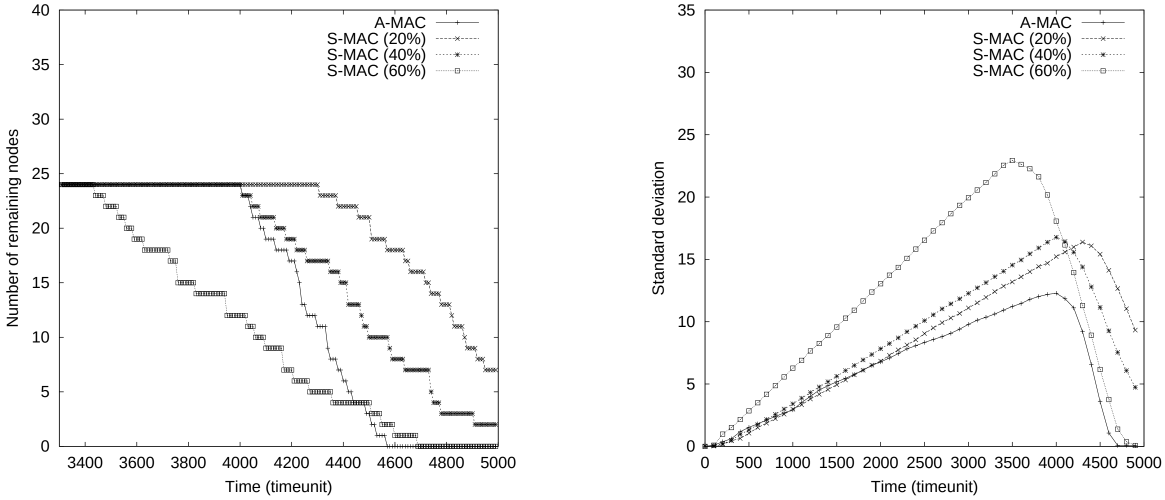
In order to evaluate the performance of our proposed scheme, we perform simulations, varying network conditions, and compare the results with those of S-MAC. In this section,



(a) Time to first node's dying

(b) End-to-end latency

Fig. 4. Performance comparison w.r.t. packet inter-arrival time (single-flow)



(a) Number of remaining nodes

(b) Standard deviation

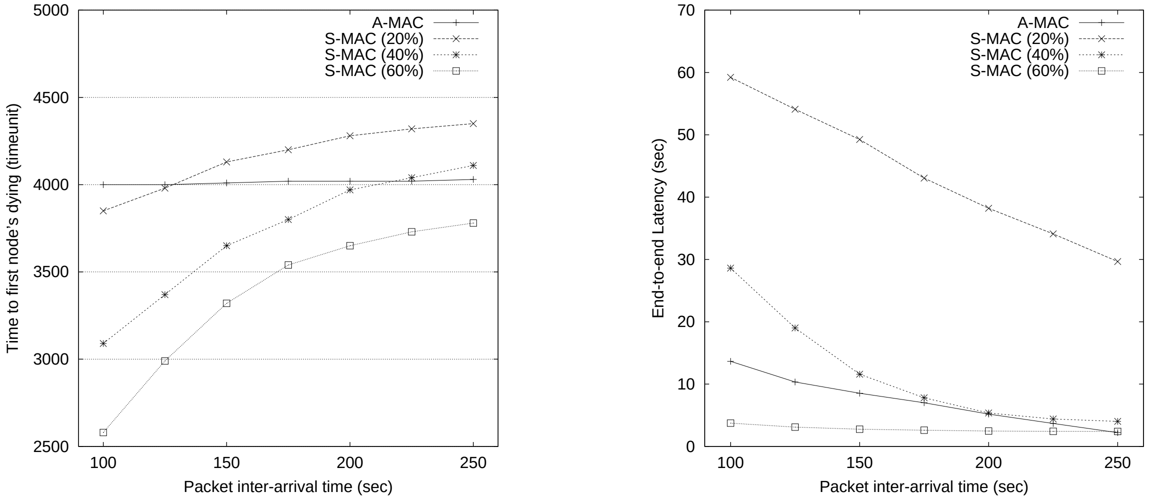
Fig. 5. Performance comparison w.r.t. time (A-MAC/S-MAC)

therefore, we consider several new metrics: the number of remaining nodes, time to first node's death, standard deviation of the energy distribution, and end-to-end latency. With these metrics, we evaluate the performance of our proposed scheme.

The simulations are conducted using NS-2 [15]. For simulations, we use 25-node topology, forming a 5 × 5 grid. The distance between adjacent nodes is 20 m and we do not assume mobility of nodes. Simulation parameters are shown in Table 1. In terms of energy consumption, we adopt the energy model in [9]. Energy-related parameters are shown in Table 2. We simulate both the single-flow and the multiple-flow cases, varying the packet inter-arrival time. We compare the results with S-MAC with varying duty cycles. During the simulations, routing metric of Max-Avg algorithm is used by default.

A. Simulation Results

1) Single-flow network: In the single-flow network, the source node is located in the top-left corner and the sink in the bottom-right, which implies at least four hops from source to sink. Fig. 4(a) exhibits time to first node's dying versus the packet inter-arrival time. A-MAC prevents network partitioning or sensing hole's occurrence until the pre-configured lifetime, 4000, regardless of the traffic load. Furthermore, the first node dies shortly after the configured lifetime. On the other hand, in S-MAC, time to first node's dying relies on both the traffic load and the duty cycle. Only S-MAC with 20% duty cycle maintains time to first node's death above 4000. End-to-end latency under the same condition is shown



(a) Time to first node's death

(b) End-to-end latency

Fig. 6. Performance comparison w.r.t. packet inter-arrival time (multiple-flows)

in Fig. 4(b). S-MAC with the high duty cycle, e.g., 60%, shows lower delay than A-MAC overall; however, it does not satisfy the pre-configured network lifetime by large margin. Likewise, S-MAC with the low duty cycle, e.g., 20%, satisfies the pre-configured lifetime, but it shows much higher latency than A-MAC. In medium and low traffic load, end-to-end latency of S-MAC is mostly dependent on the duty cycle, not on the traffic load, whereas in A-MAC, the latency diminishes as the traffic load decreases, because residual energy is utilized to reduce delay.

Fig. 5(a) plots the number of remaining nodes as time goes on, where packets are transmitted every five seconds. Here, the number next to the MAC protocol name indicates time to first node's death. A-MAC shows the steepest curve of decreasing remaining nodes, which implies almost every node dies around the end of the pre-configured network lifetime. It also signifies A-MAC has the fairest distribution of energy consumption as shown in Fig. 5(b).

2) Multiple-flow network: In the multiple-flow case, every node except the sink periodically transmits packets to the sink, which reflects the monitoring application scenario. Fig. 6(a) plots time to first node's dying with respect to the packet arrival time. The result is quite similar to that of the single-flow network. A-MAC shows almost constant distribution of time to first node's death slightly above the pre-configured lifetime, while the performance of S-MAC depends on the traffic load. This figure is also related to Fig. 6(b), which shows the end-to-end delay of A-MAC and S-MAC. S-MAC with the 60% duty cycle satisfies the required network lifetime only with the light traffic load. In such cases, A-MAC shows lower delay than S-MAC. Also, A-MAC always outperforms S-MAC with 40% or lower duty cycles in terms of delay.

3) Performace versus routing metrics: In order to evaluate the performance of A-MAC with regard to routing algorithms, we apply three routing metrics: minimum hop count (Min-Hop), Max-Min algorithm, and Max-Avg algorithm. The simulation is conducted with single-flow network with five seconds of packet inter-arrival time. Figs. 7(a) and 7(b) show time to first node's death and end-to-end latency with regard to the routing metrics. All the metrics achieve network partitioning time over the pre-configured network lifetime, 4000. However, Max-Avg shows the lowest latency than other two routing metrics.

VI. CONCLUSION

We present a novel MAC (A-MAC) protocol for WSNs. It reduces the end-to-end delay while guaranteeing the pre-configured network lifetime, using the adaptive duty cycle mechanism based on the energy consumption rate. It also achieves fairness in terms of energy consumption distribution. Simulation results reveal that A-MAC meets the network lifetime requirements while showing substantially less delay than S-MAC. Furthermore, our proposed scheme does not require careful configuration of the duty cycle of each node, which is a crucial job in S-MAC.

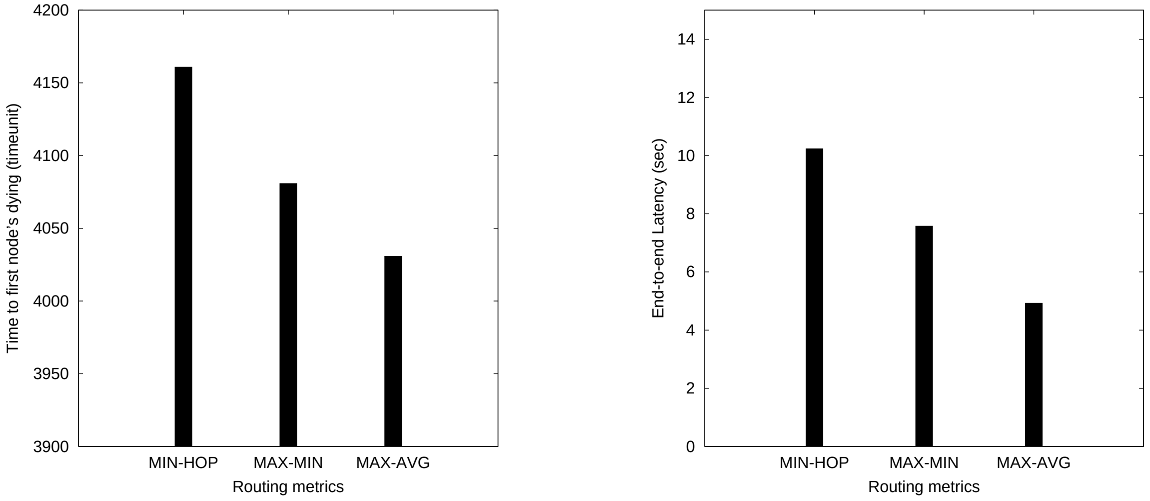
ACKNOWLEDGMENT

This work was supported in part by the Brain Korea 21 project of the Ministry of Education, and in part by grant No. R01-2004-000-10372-0 from the Basic Research Program of the Korea Science & Engineering Foundation, 2005, Korea.

REFERENCES

[1] Wei Ye, John Heidemann and Deborah Estrin, "Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks," IEEE/ACM Transactions on Networking, Vol. 12, N.3, June 2004.

[2] Tijs van Dam and Koen Lamgemdoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," Proc. of ACM SenSys 2003.



(a) Time to first node's death

(b) End-to-end latency

Fig. 7. Performance comparison w.r.t. routing metrics

[3] Tao Zheng, Sridhar Radhakrishnan and Venkatesh Sarangan, "PMAC: An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," Proc. of IEEE IPDPS 2005.

[4] Gang Lu, Bhaskar Krishnamachari and Cauligi S. Raghavendra, "An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks," Proc. of IEEE IPDPS 2004.

[5] W. R. Heinzelma, A. Chandrakasan and H. Balakrishnan, "Energy-efficient Communication Protocols for Wireless Microsensor Network," Proc. of Hawaii Int. Conf. Systems Sciences, Jan. 2000.

[6] Matthias Ringwald and Kay Romer, "BitMAC: A Deterministic, Collision-Free, and Robust MAC Protocol for Sensor Networks," Proc. of EWSN 2005.

[7] J. Pollastre, J. Hill and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," Proc. of ACM Sensys 2004.

[8] Krishna Balacandran, Joseph H. Kang and Wing Cheong Lau, "Adaptive Sleeping and Awakening Protocol (ASAP) for Energy Efficient Adhoc Sensor Networks," Proc. of IEEE ICC 2005.

[9] M. Stemm and R. H. Katz, "Measuring and Reducing Energy Consumption of Network Interfaces in Hand-Held Devices," IEICE Transactions on Communication, vol. E80-B, No. 8, pp. 1125-1131, Aug. 1997.

[10] Chao Gui and Prasant Mohapatra, "Power Conservation and Quality of Surveillance in Target Tracking Sensor Networks," Proc. of ACM MobiCom 2004.

[11] S. Singh, M. Woo and C. S. Raghavendra, "Power Aware Routing in Mobile Ad Hoc Networks," Proc. of ACM MobiCom 1998.

[12] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," Proc. of Hawaii International Conference on System Sciences, Jan. 2000.

[13] C. Schurgers and M. B. Srivastava, "Energy Efficient Routing in Wireless Sensor Networks," Proc. of IEEE MILCOM 2001.

[14] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann and F. Silva, "Directed Diffusion for Wireless Sensor Networking," IEEE/ACM Transactions on Networking, Vol. 11, N. 1, Feb. 2003.

[15] "Network simulator," http://www.isi.edu/nsnam/ns.