Research Article

An Energy and Latency Aware WSN MAC Protocol for Bidirectional Traffic in Data Collection

Ji-Yeon Min, Sang-Hwa Chung, and Yuvin Ha

Pusan National University, Busan 609-735, Republic of Korea

Correspondence should be addressed to Sang-Hwa Chung; shchung@pusan.ac.kr

Received 13 April 2014; Revised 21 August 2014; Accepted 15 December 2014

Academic Editor: Amiya Nayak

Copyright © 2015 Ji-Yeon Min et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a new Energy and Latency Aware MAC (ELA-MAC) protocol that provides high energy-efficiency and low latency for data collection applications in wireless sensor networks. In previous research, B-MAC+ avoids the receiver node’s preamble overhearing of B-MAC with short preamble packets including only the destination address and the preamble packet number. However, like B-MAC, B-MAC+ has the sender node’s preamble overhead and communication latency. XY-MAC tries to solve these problems with inserting an early ACK between preamble packets. However, with the independence of each node’s wake-up point, the receiver node’s idle listening duration of XY-MAC is always longer than that of B-MAC+. ELA-MAC spreads collection-request without using an early ACK and delivers collection-response using an early ACK with considering the traffic types such as broadcasting and convergecasting. In this way, energy consumption on both downlink and uplink can be reduced. Besides, the time for collecting data can be shortened on uplink. With the experimental parameter obtained by the energy consumption analysis, the experiment was performed on a real sensor network test bed with various topologies. We obtained the result that ELA-MAC improves the energy efficiency by 12% and 37% compared to B-MAC+ and XY-MAC, respectively, and data collection time by 28% compared to B-MAC+.

1. Introduction

A wireless sensor network consists of several sensor nodes that use constrained energy resources. These nodes need to guarantee the lifetime of the battery for at least a few months or more than a year to overcome the difficulty of recharging or replacing a discharged battery. This fact emphasizes the importance of effective management of energy consumption [1].

The design requirement for a low-power network has been an important issue, which has led to repeated research in media access control (MAC) protocols based on duty cycle [2]. These MAC protocols are classified into synchronous (sync) or asynchronous (async) based on whether the nodes are synchronized in the network, such as S-MAC [3] for sync and B-MAC [4] for async.

A synchronous MAC protocol like S-MAC uses sync-packets, which synchronize operating time for all elements. An asynchronous MAC protocol like B-MAC uses preamble sampling, which synchronizes only for nodes involved in the communication. This technique has less overhead compared to methods exchanging sync-packets. However, the length of the preamble should always be longer than the sleep duration for synchronization between nodes. Thus, it makes neighbor nodes overhear, although they do not participate in the communication.

B-MAC+ [5] addresses this preamble overhearing problem by applying the concept of a short preamble packet including the address of the destination node and the packet number. However, like B-MAC, it causes preamble overhead when the sender node transmits short preamble packets consecutively for a longer time than the sleep duration before sending the actual data to the destination node.

Between preamble packets, X-MAC [6] inserts an early ACK duration for the destination node to acknowledge the preamble packet and announce it to the sender node. However, the receiver node has to maintain an idle listening duration longer than the length of the early ACK duration.
to acknowledge the preamble packet. This extends the wake-up waiting time to the early ACK duration, compared to B-MAC+.

XY-MAC [7] adapts the early termination technique to address this issue. Due to the early ACK duration, a reserved section between the sender node and the destination node, the sender node can assess the channel condition and determine whether the destination node transmits an early ACK packet or not, through clear channel assessment (CCA), in a short period. Therefore, this technique reduces the idle listening time of the receiver node by altering the receiving time of the early ACK packet to the CCA operating time.

Each protocol proposes a technique for energy efficiency by minimizing energy consumption in the sender node or the receiver node in one-to-one communication. However, additional considerations are needed in practical use. Broadcasting to sensor nodes and converging casting to sink nodes are the major types of communication in real environments, rather than one-to-one communication. With various usages of wireless sensor networks, from context-awareness to sensing events, the demand for studying the designs of MAC protocol is on the rise, considering several factors like energy efficiency and differences in requirements for each application.

D-MAC [8] is a well-known protocol that provides high energy-efficiency and low latency for applications in sending events. Each node wakes up in turn by a given depth in a data-gathering tree and receives data from child nodes before sending its own data to its parent node. Called as staggered schedule, this is depicted in Figure 1.

In contrast, communications for applications in data collection consist of periodic requests from sink node and responses from sensor nodes, as depicted in Figure 2. This requires improved design in the MAC layer, considering two-way communication traffic, rather than applications in sending events, which is only capable of one-way communication traffic toward the sink node.

In a data collection application, it is hard to reduce both energy consumption and the time for collecting data with the existing MAC protocol such as B-MAC+ or XY-MAC which does not consider the main characteristics of two-way communication traffic. Therefore, this paper proposes what we call the ELA-MAC protocol, providing high energy-efficiency and low latency in a data collection application. ELA-MAC can determine whether an early ACK is used in any operating point according to both the direction of communication traffic and the prediction availability of communication traffic.

Following are the contents of this paper. Section 2 depicts previous research on asynchronous MAC protocols. Section 3 depicts the performance comparison of existing protocols. In Section 4, we propose the ELA-MAC protocol. Section 5 depicts the performance of the ELA-MAC protocol. In Section 6, we draw a conclusion and suggest future work.

2. Related Work

Most of the MAC protocols of wireless sensor networks are relying on duty cycle. A duty cycle is a technique to maximize energy efficiency, classifying radio transceiver modes into active and sleep, which operate alternately. The transceiver remains in active mode when data transmission or reception is needed; otherwise, it is in sleep mode to save the energy.

Previous researches could be divided into synchronous and asynchronous methods. Synchronous duty cycle MAC protocols provide energy efficiency using scheduled active period. However, they are not suitable for dense network, because every node will attempt communication simultaneously. This could cause severe collision. Furthermore, if traffic is intermittently generated, the synchronization requires additional traffic. Thus, it is not suitable to low data rate network.

In this paper, synchronous duty cycle MAC protocols are not discussed because of the following reasons. First, typical data collection applications belong to low data rate network in which a synchronous MAC protocol has synchronization overhead. Second, synchronous MAC protocols could suffer serious collision during the data collection-response period because the collection-response generates many-to-one traffic.

While all nodes are synchronized for operation in synchronous MAC protocols, each node has a different wake-up point in asynchronous MAC protocols. Thus, media access of sender nodes could be distributed. Naturally, it does not have any synchronization overhead. However, asynchronous MAC protocols require efficient wake-up techniques because destination nodes wait in wake-up mode until actual data receptions are done.

2.1. B-MAC. A low data rate in a wireless sensor network brings unnecessary energy consumption in sensor nodes from maintaining a long idle listening time due to the intermittent communication. Because of this, existing asynchronous MAC protocols are focusing on saving energy of idle listening.
B-MAC is a representative asynchronous MAC protocol that adopts preamble sampling to reduce idle listening time. As depicted in Figure 3, the node that would send data needs to transmit a longer preamble than the length of the sleep duration. The other nodes check conditions on the radio channel for a while at their wake-up point to make note of the preamble. When the node wakes up and notes the preamble, it waits to receive the actual data. The sender node sends the actual data, including the address of the destination node, after transmitting a long preamble. Then, the destination node can successfully receive the actual data and complete the communication.

By applying preamble sampling, communication is even possible when the sender node does not synchronize with neighbor nodes. However, a longer preamble than the length of the sleep duration makes all neighbors of the sender node note the preamble at their wake-up point, regardless of the fact whether they are the destination node or not. In addition, they must receive the long preamble for receiving the actual data to check the destination address. This can cause the receiver node to use the radio unnecessarily, wasting energy.

2.2. B-MAC+. To address the preamble overhearing problem in B-MAC, B-MAC+ uses a short preamble packet including the address of the destination node and the preamble packet number. As depicted in Figure 4, the node that wants to send data transmits short preamble packets consecutively for a longer time than the length of the sleep duration. When a node wakes up and detects a preamble packet, it checks the address of the destination node included in the preamble packet. If it is not the destination node, it goes back into sleep mode, or if it is the destination node, it returns to sleep mode until it receives the actual data using the packet number included in the short preamble packet.

B-MAC+ tries to solve the preamble overhearing problem using short preamble packets including the address of the destination node and the preamble packet number. As with B-MAC, a sender node has to transmit a longer preamble than length of the sleep duration of receiver nodes. It also causes the sender node to overuse energy for the radio.

2.3. X-MAC. As depicted in Figure 5, X-MAC inserts an early ACK duration between preamble packets, enabling the destination node to wake up and declare its status to the sender by transmitting an early ACK packet. Then the sender stops transmitting the preamble packet and sends the actual data at the point where the destination node responds to the preamble packet.

X-MAC tries to reduce the long preamble overhead in the sender node with the early ACK, but this increases idle listening duration at each wake-up point for the receiver node. This occurs from the independence of each node’s wake-up point in the asynchronous MAC protocol. To make receiver nodes that wake up in the early ACK duration of the sender node detect the preamble packet from the sender node, the receiver node should wait in the idle listening state for a longer time than the early ACK duration. The increase in idle listening duration causes energy waste in the receiver node at every wake-up point, until the lifetime of the battery expires.

2.4. XY-MAC. XY-MAC applies the early termination technique, minimizing the early ACK duration in the sender node to reduce the idle listening duration in the receiver node. All sender nodes do not need to wait to receive the whole early ACK packet, because the early ACK duration is a reserved section between sender node and destination node. The sender node waits while it can determine whether the destination node sends an early ACK packet or not through CCA.

As depicted in Figure 6, after transmitting a preamble packet, the sender node determines whether the destination node transmits an early ACK packet only during the CCA duration. To do this, when the early ACK duration starts, the sender node detects the channel. If the channel is idle, the sender node transmits the next preamble packet immediately, because the destination node has not transmitted an early ACK packet. If the channel is busy, the sender node waits for an early ACK and receives it because the destination node has transmitted an early ACK packet.
Thus, early termination can reduce the idle listening time and energy consumption of the receiver node by shortening the early ACK duration, compared to X-MAC.

3. Comparison of Existing B-MAC+ and XY-MAC

In this section, we perform a simple experiment to compare B-MAC+ with XY-MAC in terms of communication time and energy consumption.

A sensor node designed and manufactured in the laboratory was used for the experiment. We applied the same frame structure of the preamble packet presented by the XY-MAC and used a symbol rate of 55.5 Kbps. In the previous work of XY-MAC [7], the CCA duration for early termination at the sender node and the idle listening duration at the receiver node were set to 2 ms and 5 ms, respectively. Because of hardware limitations [9], the CCA duration requires at least 2 ms. To detect the preamble packet, the idle listening duration of XY-MAC should be larger than early ACK duration. Likewise, because of hardware limitations such as RSSI response time and synthesizer calibration, the idle listening duration at the receiver node requires 5 ms.

Without using early ACK, the receiver node only needs to maintain its idle listening duration during the CCA duration to recognize a preamble packet. Thus, for B-MAC+, we applied a time of 2 ms, which is equal to the CCA duration in the XY-MAC for the idle listening duration at the receiver node.

3.1. Communication Time. Figures 7 and 8 indicate the communication times under each protocol. We defined communication time as the time it takes for the receiver node to receive data from the sender node. Communication time was measured from the start of sending data at the sender node.

Figure 7 indicates the communication time under each protocol according to the sleep duration when the sender node and the destination node communicate in a single hop. For B-MAC+, the communication is always completed after a time period longer than the sleep duration. On the other hand, in case of XY-MAC, the communication is completed, on average, within a time period shorter than the sleep duration. When using early ACK (i.e., XY-MAC), communication time can be shorter than the sleep duration.

Figure 8 indicates the communication time under each protocol according to the sleep duration and the number of hops when the sender node and the destination node communicate in multihops. For XY-MAC, the communication time is increased by a time period shorter than the sleep duration, on average, even though the number of hops increases. On the other hand, in case of B-MAC+, the communication time is always increased by a time period longer than the sleep duration, as the number of hops increases. Without using early ACK (i.e., B-MAC+), as a result of Figure 7, the communication is always completed after a time period longer than the sleep duration at every hop.

3.2. Communication Energy Consumption. Energy consumption on the receiver node is almost always due to idle listening at every wake-up point. Thus, energy consumption on the receiver node under each protocol can be evaluated by the idle listening duration. XY-MAC and B-MAC+ maintain idle listening for 5 ms and 2 ms, respectively. Thus, when traffic does not exist, B-MAC+ has about 2.5 times higher performance than XY-MAC in terms of energy efficiency of receiver nodes. In other words, if there is no traffic, not using early ACK is more suitable because of short idle listening.

On the other hand, energy consumption on the sender node is almost totally due to using the radio while transmitting the preamble packet or recognizing the early ACK.
packet. Assuming that the energy to transmit a preamble packet and the energy to recognize an early ACK packet are the same, energy consumption by the sender node under each protocol can be evaluated by the communication time. Through the experiment performed in Section 3.1, the communication time under XY-MAC is about half of B-MAC+. Thus, in one communication, XY-MAC has an energy efficiency about twice higher than that of B-MAC+. In other words, when nodes communicate with each other, using early ACK is more suitable because of early termination.

However, energy consumption on the sender node actually occurs only during communication between the sender node and the destination node, and energy consumption by the receiver node occurs continuously at every wake-up point. Therefore, in order to compare energy consumed from communication under each protocol, it is necessary to consider energy consumption by continuous communication rather than just from one communication.

A single experiment where the sender node requests data and the destination node responds to it was carried out repeatedly every 30 s to measure performance. The destination node operated until its own remaining energy was only 5%, and the experiment was carried out until data was not received by the sender node. For comparison of each protocol, the experiment was carried out by fixing 100 ms of the sleep duration.

Figure 9 indicates energy consumption under each protocol. From the result of the experiment, the lifetime of B-MAC+ and XY-MAC is 14,646 rounds and 11,891 rounds, respectively, and there is a difference of about 23%.

Using early ACK, in one communication, it is possible to reduce energy consumption at the sender node and shorten the communication time. However, an increase of the idle listening duration caused by early ACK affects receiver nodes at every wake-up point. It leads to poor performance in terms of overall energy efficiency.

4. ELA-MAC Protocol

4.1. Analysis of Communication Pattern of Data Collection Application. The purpose of a data collection application is to collect data from many sensor nodes. Therefore, communication by the data collection application can be classified as a request by the sink node and a response by the sensor node.

The collection-request from the sink node is broadcasted on downlink to all sensor nodes in a network distributed over a wide area. Broadcasting can be persisted by multihops due to the size of the network. On the other hand, the collection-responses from all sensor nodes are convergecasted on uplink towards the sink node. Convergecasting can be also persisted by multihops due to the size of the network. In a general convergecasting environment, it is advantageous that a node sends its own data together with data of its child nodes along a preset path to the sink node in terms of reducing energy consumption by the sensor node in a data transfer [10–12]. At this time, the spanning tree rooted at the sink node may be used as the connection between sensor nodes, as described in Figure 10. More detailed information is beyond the scope of this paper.

Therefore, the direction of communication traffic on uplink and on downlink is the major transmission pattern of a data collection application. These directions of communication traffic generate a new transmission pattern for the data collection application: prediction of the communication traffic.

The sensor node cannot predict the occurrence of communication traffic on downlink before the sink node requests data. However, after the sink node requests data, the sensor node can predict the occurrence of communication traffic on uplink. This is because the sensor node can determine for itself that it lies on multihop routing path from the leaf node on uplink if the network is connected via multiple hops, not a single hop, and the sensor node is not a leaf node.

4.2. ELA-MAC Protocol Considering Communication Pattern of Data Collection Application. In this section, we propose the ELA-MAC protocol which is proper to a data collection application. ELA-MAC considers the main characteristics of two-way communication traffic in a data collection application.

First, while waiting for the collection-request, it is not possible for a sensor node to predict when communication traffic on downlink is generated. Thus, the sensor node must wait for the collection-request with its own duty cycle. In low data rate networks, using the radio for the idle listening at
4.3. Process of ELA-MAC Protocol. Figure 12 is a frame structure of XY-MAC [7]. In the existing XY-MAC, PacketType is used to distinguish the preamble packet and the early ACK packet. However, under the ELA-MAC, the use of PacketType changes to distinguish the four packets: the preamble packet for broadcasting, the preamble packet for convergecasting, the early ACK packet, and the data packet. The receiver nodes can determine their behavior from PacketType.

Then, we will explain the operation process of the ELA-MAC in detail. Algorithms 1, 2, 3, 4, and 5 indicate the pseudocode for the algorithms of the operation process of the sink node and the sensor node, respectively. The operation process is largely divided into the broadcast phase and the convergecast phase, and each node executes these phases repeatedly until they run out.

(i) Operation Process of Sink Node

(a) Broadcast Phase. The sink node transmits preamble packets for broadcasting without using an early ACK and spreads the collection-request to the child nodes in order to request sensor data.

(b) Convergecast Phase. The sink node waits for the collection-response from the child nodes. If the sink node recognizes a preamble packet for convergecasting, it determines whether it is the destination node from the destination address contained in the preamble packet. If the sink node is the destination node, by transmitting an early ACK packet to the sender node, it receives the actual data.

(ii) Operation Process of Sensor Node

(a) Broadcast Phase. First, the general node and the leaf node equally perform the duty cycle without using an early ACK in order to wait for the collection-request. At this time, if the sensor node recognizes a preamble packet for broadcasting, it maintains the sleep state for a short time and wakes up again to receive the actual data according to the packet number contained in the preamble packet. If the sensor node is not the leaf node, it transmits preamble packets for broadcasting without using an early ACK and request data packet to the child nodes in order to spread the collection-request.

(b) Convergecast Phase. If the sensor node is the leaf node that does not have child nodes, it immediately transmits preamble packets for convergecasting with using early ACK and response data packet to the parent node. The general node that is not the leaf node performs the duty cycle with using an early ACK in order to wait for the collection-response from the child nodes. At this time, if the sensor
(1) \textbf{procedure} Sink(sleepDuration, collectionIntervalDuration)
(2) \hspace{1em} \textbf{while} true \textbf{do}
(3) \hspace{2em} \textbf{while} sleepDuration \textbf{do}
(4) \hspace{3em} send preamble packet without early ack
(5) \hspace{2em} \textbf{end while}
(6) \hspace{1em} send request data packet
(7) \hspace{1em} collectionTimer = collectionIntervalDuration
(8) \hspace{1em} phase = CONVERGCAST\_PHASE
(9) \hspace{1em} \textbf{while} collectionTimer \textbf{do}
(10) \hspace{2em} \textbf{if} receive convergecasted preamble packet \textbf{then}
(11) \hspace{3em} \textbf{if} destination == myaddress \textbf{then}
(12) \hspace{4em} send early ack packet
(13) \hspace{3em} \textbf{else}
(14) \hspace{4em} receive response data packet
(15) \hspace{3em} \textbf{end if}
(16) \hspace{2em} \textbf{end if}
(17) \hspace{1em} phase = BROADCAST\_PHASE
(18) \hspace{1em} \textbf{end while}
(19) \hspace{1em} \textbf{end procedure}

\textbf{Algorithm 1:} The sink node algorithm.

(1) \textbf{procedure} Sensor(sleepDuration, collectionIntervalDuration)
(2) \hspace{1em} phase = BROADCAST\_PHASE
(3) \hspace{1em} \textbf{while} true \textbf{do}
(4) \hspace{2em} \textbf{if} phase == BROADCAST\_PHASE \textbf{then}
(5) \hspace{3em} \textbf{while} sleepDuration \textbf{do}
(6) \hspace{4em} RF OFF
(7) \hspace{3em} \textbf{end while}
(8) \hspace{2em} \textbf{else}
(9) \hspace{3em} RF ON
(10) \hspace{3em} \textbf{while} idle listen duration without early ack \textbf{do}
(11) \hspace{4em} if receive broadcasted preamble packet \textbf{then}
(12) \hspace{5em} ProcessingBroadcastPreamble()
(13) \hspace{4em} \textbf{end if}
(14) \hspace{3em} \textbf{end if}
(15) \hspace{2em} \textbf{end if}
(16) \hspace{1em} \textbf{if} phase == CONVERGCAST\_PHASE \textbf{then}
(17) \hspace{2em} \textbf{while} collectionTimer \textbf{do}
(18) \hspace{3em} if not leaf node \textbf{then}
(19) \hspace{4em} RF OFF
(20) \hspace{4em} \textbf{end while}
(21) \hspace{3em} IdleListeningUsingEarlyACK()
(22) \hspace{3em} \textbf{end if}
(23) \hspace{2em} \textbf{else}
(24) \hspace{3em} SensorDataResponse()
(25) \hspace{2em} \textbf{end if}
(26) \hspace{2em} \textbf{end if}
(27) \hspace{1em} phase = BROADCAST\_PHASE
(28) \hspace{1em} \textbf{end if}
(29) \hspace{1em} \textbf{end while}
(30) \hspace{1em} \textbf{end procedure}

\textbf{Algorithm 2:} The sensor node algorithm.
node recognizes a preamble packet for convergecasting, it determines whether it is the destination node from the destination address contained in the preamble packet. If it is the destination node, by transmitting an early ACK packet to the sender node, it receives the actual data. Then, the sensor node transmits preamble packets for convergecasting with using early ACK and response data packet formed by merging data of the child nodes to deliver the collection-response to the parent node.

A sensor node operates without early ACK before receiving a collection-request (i.e., broadcast phase). When the sensor node forwards the collection-request to its child nodes after receiving the request from the parent node, it changes its own operation mode to convergecast phase. At the same time, the sensor node sets its timer using the time information included in the request packet. When the timer expires, its operation mode returns to broadcast phase again.

The time duration implies a valid period of collection-request and response. Thus, this duration must be long enough to ensure that the sink node can receive all collection-responses. However, determining an optimum duration is beyond the scope of the paper. In this paper, we assume that the given time duration for a collection operation is long enough.

To simultaneously terminate the timer of all sensor nodes in network, received time information should be compensated, because the received time information is different between the parent node and the child node. Figure 13 describes the time interval between nodes. The time interval between parent and child nodes can be calculated as follows (see Table 1):

$$T_i = T_{pream} + D_{trans} + D_{prop}.$$  (1)

As depicted in Algorithm 4, a calculated time interval should be deducted from the received time information before forwarding it to its child node. The preamble duration and transmission delay are deterministic value while the propagation delay can vary according to distance between nodes. However, the propagation delay can be fixed to the maximum value obtained from experiment, because the variation of the propagation delay according to the distance is very small.
Algorithm 5: The local function of the sensor node “SensorDataResponse”.

Now, we suppose the case that the sink node collects data from node B, via node A, as depicted in Figure 13. First, the sink node changes its operation mode to convergecast phase after transmitting a collection-request to node A. At the same time, the sink node set the timer using the given time information. Likewise, node A that received collection-request forwards the request to node B and changes its operation mode and then sets the timer.

Now, we consider an exceptional case. When node B fails to receive the request from node A, another new collection-request is broadcasted by the sink node. At this time, we can consider the cases shown below.

(i) Before A’s Timer Expires. In this case, node A would operate under convergecast phase that uses early ACK, because the timer is not expired yet. Note that the idle listening duration of convergecast phase is longer than the ones of broadcast phase, 5 ms and 2 ms, respectively. Furthermore, because the two phases use the same preamble packet structure of Figure 12, node A that operates under convergecast phase can receive the preamble packet of the sink node that operates under broadcast phase. After that, using PacketType field of the preamble packet, node A can identify the MAC mode of the sender node. It enables that node A wakes up at the exact point of data transmission and receives the new request. This scenario is depicted in Figure 14. In other words, there is no problem with communication between a sender node that does not use early ACK and a receiver node that uses early ACK. If a sensor node receives another new request before timer expiration, it updates its timer using the latest information.

(ii) After A’s Timer Expires. In this case, node A would change the operation mode to broadcast phase because of the timer expiration. Thus, there is no problem.

4.4. Analysis of Energy Consumption by Sensor Node. In this section, we analyze energy consumption by the sensor node operating under ELA-MAC.
All sensor nodes repeatedly execute processes (1) to (6) below. (However, the leaf node does not require steps (4) and (5).)

1. Wait for the collection-request.
2. Receive collection-request.
3. Send collection-request.
4. Wait for the collection-response.
5. Receive collection-response.
6. Send collection-response.

Table 2 indicates variables used to build a mathematical formula of energy consumption by the sensor node. The energy consumption of each step can be represented by the formulas as follows.

(1) Waiting for the Collection-Request. Energy consumption from waiting for the collection-request is due to idle listening based on B-MAC+ at every wake-up point. Since the idle listening duration based on B-MAC+ is fixed at 2 ms, energy consumption will be different according to the number of wake-up points. Because the number of wake-up points is determined by the collection interval duration and the sleep duration, energy consumption from waiting for the collection-request may be expressed as follows:

\[
E_{w,\text{req}} = D_{\text{b,mac+}} \times C_{\text{b,mac+}} \times P_{\text{rx}}
\]

\[= D_{\text{b,mac+}} \times \frac{D_{\text{collection}}}{D_{\text{sleep}}} \times P_{\text{rx}} \quad (2)
\]

(2) Receiving Collection-Request. Energy consumption from receiving the collection-request is due to the reception of a preamble packet based on B-MAC+ and a request data packet. The length of all packets is the same; energy consumption from receiving the collection-request may be expressed as follows:

\[
E_{r,\text{req}} = 2 \times D_{\text{packet}} \times P_{\text{rx}}
\]

(3) Sending Collection-Request. Energy consumption from sending the collection-request is due to the transmission of the preamble packets based on B-MAC+ and a request data packet. Because it is known that the communication time under B-MAC+ is about the sleep duration through the experiment performed in Section 3.1, energy consumption

| Variable       | Description                        | Unit   |
|----------------|------------------------------------|--------|
| \(D_{\text{collection}}\) | Collection interval duration | (s)    |
| \(D_{\text{sleep}}\)      | Sleep duration                       | (s)    |
| \(D_{\text{packet}}\)     | 15-byte packet duration             | (s)    |
| \(D_{\text{b,mac+}}\)     | B-MAC+ idle listening duration      | (s)    |
| \(D_{\text{xy,mac}}\)     | XY-MAC idle listening duration      | (s)    |
| \(D_{\text{early,ack}}\)  | Early AK duration                   | (s)    |
| \(C_{\text{b,mac+}}\)     | B-MAC+ idle listening count         |        |
| \(C_{\text{xy,mac}}\)     | XY-MAC idle listening count         |        |
| \(P_{\text{tx}}\)         | Power for transmission              | (W)    |
| \(P_{\text{rx}}\)         | Power for reception                 | (W)    |
| \(E_{w,\text{req}}\)      | Energy to wait for data request     | (J)    |
| \(E_{s,\text{req}}\)      | Energy to receive data request      | (J)    |
| \(E_{s,\text{resp}}\)     | Energy to send data request         | (J)    |
| \(E_{s,\text{resp}}\)     | Energy to receive data response     | (J)    |
| \(E_{\text{once}}\)       | Energy for only one data collection | (J)    |
from sending the collection-request may be expressed as follows:
\[ E_{s,\text{req}} = (D_{\text{sleep}} + D_{\text{packet}}) \times P_{\text{tx}}. \]  

(4) **Waiting for Collection-Response.** Energy consumption from waiting for the collection-response is due to idle listening based on XY-MAC at every wake-up point. Since the idle listening duration based on XY-MAC is fixed at 5 ms, energy consumption will be different according to the number of wake-up points. Because the number of wake-up points is determined by the number of hops from the leaf node and the network environment, energy consumption from waiting for the collection-response may be expressed as follows:
\[ E_{w,\text{resp}} = D_{\text{xy,mac}} \times C_{\text{xy,mac}} \times P_{\text{rx}}. \]  

(5) **Receiving Collection-Response.** Energy consumption from receiving the collection-response is due to the reception of a preamble packet based on XY-MAC, the transmission of an early ACK packet, and the reception of a response data packet. The length of all packets is the same; energy consumption from receiving the collection-response may be expressed as follows:
\[ E_{r,\text{resp}} = D_{\text{packet}} \times ((2 \times P_{r_x}) + P_{\text{tx}}). \]  

(6) **Sending Collection-Response.** Energy consumption from sending the collection-response is due to the transmission of the preamble packets based on XY-MAC, CCA for early termination, the reception of an early ACK packet, and the transmission of a response data packet. Because it is known that the communication time under XY-MAC is about half the sleep duration through the experiment performed in Section 3.1, energy consumption from sending the collection-response may be expressed as follows:
\[ E_{s,\text{resp}} = \left( \frac{D_{\text{sleep}}}{2 \times (D_{\text{packet}} + D_{\text{early,ack}})} \right) \times \left( (D_{\text{packet}} \times P_{\text{tx}}) + (D_{\text{early,ack}} \times P_{\text{rx}}) \right) + (D_{\text{packet}} \times (P_{\text{rx}} + P_{\text{tx}})). \]  

The energy consumption of the whole process can be represented as the sum of the energy consumption of each step. Accordingly, the entire energy consumption of the sensor node when data collection occurs only once can be expressed as follows:
\[ E_{\text{once}} = E_{w,\text{req}} + E_{s,\text{req}} + E_{w,\text{resp}} + E_{r,\text{resp}} + E_{s,\text{resp}}. \]  

Some communication parameters (\(D_{\text{packet}}, D_{\text{b,mac}}, D_{\text{xy,mac}}, \text{and } D_{\text{early,ack}}\)) are same with parameters of XY-MAC [7]. Thus, if the hardware platform is selected, energy consumption that is not affected by the values of \(D_{\text{collection}}\) and \(D_{\text{sleep}}\) can be constant as fixed values. On the other hand, the values of \(D_{\text{collection}}\) and \(D_{\text{sleep}}\) are different depending on the requirements of the application such as low power consumption, reliability, and real time. Consequently, overall energy consumption is determined by the values of \(D_{\text{collection}}\) and \(D_{\text{sleep}}\). Assuming that the data rate is fixed at 55.5 Kbps and the values of \(P_{\text{tx}}\) and \(P_{\text{rx}}\) are fixed at the same value \(P\), energy consumption that is determined by the values of \(D_{\text{collection}}\) and \(D_{\text{sleep}}\) can be expressed as follows:
\[ E_{\text{trans}} = E_{w,\text{req}} + E_{s,\text{req}} + E_{w,\text{resp}} + E_{r,\text{resp}} + E_{s,\text{resp}}. \]  

If data collection is continuously conducted during a certain time \(D\), \(E_{\text{trans}}\) can be expressed as follows:
\[ E_{\text{continuous}} = \frac{D}{D_{\text{collection}}} \times E_{\text{trans}}. \]  

**Figure 15** shows \(E_{\text{continuous}}\) when \(D_{\text{collection}}\) is from 1 hour to 10 hours.
at a large network environment. So, appropriately adjusting both $D_{\text{collection}}$ and $D_{\text{sleep}}$ is required, depending on the requirements of the application.

Figure 16 shows $E_{\text{continuous}}$ when $D_{\text{collection}}$ is a relatively short time which is from 1 second to 10 seconds. As $D_{\text{collection}}$ decreases and $D_{\text{sleep}}$ increases, energy consumption is higher. First, when $D_{\text{collection}}$ decreases, the reason why energy consumption is higher is the same as described in Figure 15. Second, when $D_{\text{sleep}}$ increases, the reason why energy consumption is higher is due to the increase in the preamble transmission duration of the sender node. With a longer sleep duration, the idle listening duration of the receiver node can decrease. At the same $D_{\text{collection}}$, as $D_{\text{sleep}}$ increases, the number of wake-up points of the receiver node decreases. However, as $D_{\text{sleep}}$ increases, the preamble transmission duration of the sender node increases. When $D_{\text{collection}}$ is a relatively short time which is from 1 second to 10 seconds, the increase in the preamble transmission duration of the sender node is greater than the decrease in the idle listening duration of the receiver node. Therefore, setting $D_{\text{sleep}}$ for a shorter time is how to improve energy efficiency in applications where data collection is frequently conducted.

For example, $D_{\text{collection}}$ is a short time in applications that continuously monitor the sensing area. In this paper, with assuming such applications, all the experiments will be performed by setting $D_{\text{collection}}$ for a time as short as possible. Also, $D_{\text{sleep}}$ will be set for both 100 ms and 200 ms in order to measure the data collection time and the data collection rate according to $D_{\text{sleep}}$. When $D_{\text{sleep}}$ is 200 ms, the data collection time is about 4000 ms with 10 sensor nodes. $D_{\text{collection}}$ must be longer than the data collection time. Thus, in all the experiments, $D_{\text{collection}}$ will be set for 5 s that is sufficiently longer than the data collection time. Also, in the energy consumption experiment, $D_{\text{sleep}}$ will be set for 100 ms that improves energy efficiency when $D_{\text{collection}}$ is 5 s.

5. Experiment and Evaluation

5.1. Experiment Environment. Figure 17 is a picture of the sensor node designed and manufactured in the laboratory for the experiment. This sensor node is equipped with a TI CC430F6137 SoC that combines a CCI101 with an MSP430. It has 32 KB of flash memory, 2 KB of RAM, and a 26 MHz clock speed. And it is supplied with 3 V from two AA batteries [13].

Table 3 indicates the communication parameters in the experiment. The packet of 15 bytes was used under both B-MAC+ and XY-MAC. Frame structure of the packet is the same as in Figure 12, and more information is described in Section 4.3. In the existing XY-MAC, CCA or frame delimiter detection (FDD) can be selected for early termination. But in this paper, CCA to be executed during a relatively short time was selected for early termination.

As described in Figures 18–20, spanning tree, linear multihops, and star topology were used for the data collection network structure. All parent nodes and child nodes were fixed arbitrarily for communications.

In the existing XY-MAC, traffic is generated intermittently; collisions will not occur at all. However, CSMA/CA must be applied, because interference can occur in the traffic types of broadcasting and convergecasting.

If backoff time is too short, failure of channel access occurs frequently because of other nodes already transmitting the preamble packets during the sleep duration, and, from this, energy consumption increases unnecessarily. On the other hand, if backoff time is too long, data collection time increases unnecessarily. Thus, backoff time was set to half the sleep duration. Furthermore, rebroadcasting was conducted only once to prevent broadcast storms.

Each node operates as described in Sections 4.2 and 4.3; all sensor nodes except the sink node were supplied with energy from the same type of battery. A single experiment where the sink node spreads the collection-request and many sensor nodes deliver the collection-response was carried out repeatedly every 5 s to measure performance. Based on the analysis in Section 4.4, the collection interval duration was
set for a time as short as possible to assume the application that continuously monitors the sensing area.

All sensor nodes operated until their own remaining energy was only 5%, and the experiment was carried out until the collection-response was not received by the sink node. In order to verify the performance of the proposed protocol in this paper, the performance of existing MAC protocol, such as B-MAC+ or XY-MAC, was compared.

5.2. Performance Evaluation

5.2.1. Data Collection Time. Figures 21–23 are indicative of the data collection time under each protocol. We defined data collection time as the time it takes for the sink node to send the collection-request and receive the collection-responses from multiple sensor nodes. Data collection time was evaluated from the start of sending the collection-request at the sink node, and the value of the performance includes a delay time due to backoff. For the data collection time comparison, in all topologies, the experiment was carried out while changing the sleep duration.

First, when operating under ELA-MAC rather than operating under B-MAC, the data collection time is short in all topologies even though the sleep duration is long. When operating under ELA-MAC, communication time at each hop can be shortened by using early ACK on uplink. On the other hand, when operating under B-MAC+, as shown in Section 3.1, because the communication completes after a time period longer than the sleep duration in every hop, the data collection time is longer than that of ELA-MAC. In the results of the experiment, the data collection time of ELA-MAC is shorter than that of B-MAC+ by about an average of 28%.

Next, when operating under XY-MAC, the data collection time is similar to operating under ELA-MAC. Because the collection-request must be broadcasted, the sender node must transmit the preamble packets during a time period longer than the sleep duration so that all neighboring nodes can recognize it. Thus, communication time cannot be shortened by using early ACK on downlink. Therefore, in the results of the experiment, the data collection time of XY-MAC is similar to that of ELA-MAC.

5.2.2. Data Collection Energy Consumption. Figure 24 indicates energy consumption under each protocol. For energy consumption comparison, in topology A, the experiment was carried out by fixing 100 ms of the sleep duration. Based on the analysis in Section 4.4, the sleep duration was set for 100 ms instead of 200 ms to improve energy efficiency.
In addition, the average value of the remaining energy in each node was calculated.

The energy efficiency of ELA-MAC is higher than B-MAC+ with a difference of about 12% of the average residual energy in each round. This difference in performance is due to the communication traffic on uplink. The preamble transmission duration of the sender node is much longer than the idle listening duration of the receiver node for communication traffic on uplink, which is reserved. Thus, when operating under ELA-MAC, it is possible to take advantage of the reduction of the preamble transmission duration of the sender node by operating with an early ACK for communication traffic on uplink.

The energy efficiency of ELA-MAC is higher than XY-MAC with a difference of about 37% of the average residual energy in each round. This difference in performance is due to the communication traffic on downlink. The idle listening duration of the receiver node is much longer than the preamble transmission duration of the sender node for communication traffic on downlink, which is not reserved. Thus, when operating under ELA-MAC, it is possible to take advantage of the reduction of the idle listening duration of the receiver node by operating without an early ACK for communication traffic on downlink.

The energy efficiency of B-MAC+ is higher than XY-MAC with a difference of about 18% of the average residual energy in each round. This difference in performance is due to communication traffic on both uplink and downlink. An early ACK can decrease the preamble transmission duration of the sender node for communication traffic on uplink. However, an early ACK also can increase the idle listening duration of the receiver node for communication traffic on downlink. Therefore, poor performance occurs from the point of view of overall energy efficiency.

Figure 25 indicates the remaining energy under each protocol. For comparison, the energy remaining in each node was measured at the 30,000th round.

The low performance for residual energy in the relay node compared to the leaf node was common to all protocols. This is due to the difference in the number of communications executed by the relay node and the leaf node in each round.
Unlike the relay node, which spreads the collection-request and delivers the collection-response, the leaf node has to deliver the collection-response only once. Therefore, energy consumption in the relay node is high compared to the leaf node in each round.

In a network connected via multiple hops instead of a single hop, the lifetime of the entire network is determined by the lifetime of the node that is one hop away from the sink node. If the node that is one hop away from the sink node does not work anymore, it is impossible to deliver the data to the sink node even though there is enough residual energy in the child node.

Figure 26 indicates a reconstituted topology structure of Figure 18 in the form of clusters on the basis of the node that is one hop away from the sink node. In such an environment, in order to extend the lifetime of the entire network, reducing the residual energy deviation at each cluster is required.

However, the number of communications generated by the relay node cannot be changed because it is fixed. Meanwhile, the number of transmissions of the preamble packet used for communication can be changed flexibly. Since it is necessary to broadcast in communication traffic on downlink, the number of transmissions of the preamble packet cannot be reduced by using early ACK. However, since it is necessary to convergecast in communication traffic on uplink, the number of transmissions of the preamble packet can be reduced by using early ACK. Thus, it is possible to reduce the residual energy deviation between the relay node and the leaf node by using early ACK.

Figure 27 indicates the residual energy deviation under each protocol. For comparison, the deviation in residual energy was calculated at each cluster. A cluster is configured based on the node that is one hop away from the sink node and includes all child nodes.

First, when operating under ELA-MAC and XY-MAC, the residual energy deviation is smaller than when operating under B-MAC+. With B-MAC+, the number of communications and the number of transmissions of the preamble packet occurring in a relay node are fixed. On the other hand, with ELA-MAC and XY-MAC, the number of transmissions of the preamble packet can be reduced by using early ACK in communication traffic on uplink. Thus, it is possible to reduce the residual energy deviation between the relay node and the leaf node by using early ACK.

Next, when operating under XY-MAC, the residual energy deviation is smaller than when operating under ELA-MAC. However, under XY-MAC, with the increase in idle listening duration from early ACK, as shown in Figure 24, in every round, the energy consumption is high. Therefore, in view of overall energy efficiency, performance is not good compared with ELA-MAC.

5.2.3. Data Collection Rate. Figure 28 indicates the data collection rate under each protocol. For the data collection rate comparison, in topology A, the experiment was carried out while changing the sleep duration. In addition, the average value of the data collection rate for up to 50,000th rounds was calculated.

First, when operating under ELA-MAC and XY-MAC, the data collection rate is higher than when operating under B-MAC+. This performance occurs because ELA-MAC and XY-MAC apply early ACK in communication traffic on uplink. With using early ACK, the sender node sends the
actual data to the destination node after receiving the early ACK packet. Thus, it is possible for the destination node to ensure reception of the actual data, because the sender node sends the actual data when the receiver node is ready to receive data. On the other hand, without using early ACK, the sender node sends the actual data to the destination node immediately after transmitting preamble packets. Thus, it is not possible for the destination node to ensure reception of the actual data if there are problems like failure of the receiver node or interference from another medium.

Next, when operating under B-MAC+, as the length of sleep duration increases, the data collection rate is lower. But when operating under ELA-MAC and XY-MAC, as the length of sleep duration increases, the data collection rate is not lower. This performance occurs because B-MAC+ does not apply early ACK in communication traffic on uplink. Without using early ACK, the probability of communication failure will increase when sleep duration is longer. On the other hand, with using early ACK, the probability of communication failure will be the same, even though sleep duration is longer.

6. Conclusion and Future Work

In this paper, we propose ELA-MAC protocol that provides high energy-efficiency and low latency for data collection in a wireless sensor network. For performance verification of the proposed protocol, we constructed a wireless sensor network test bed and compared scenarios when operating under B-MAC+ and XY-MAC.

First, the data collection time of ELA-MAC is shorter than that of B-MAC+ by about an average of 28% because communication time can be shortened by using early ACK for communication traffic on uplink. But the data collection time of XY-MAC is similar to that of ELA-MAC because communication time cannot be shortened by using early ACK for communication traffic on downlink.

Next, energy consumption under ELA-MAC is higher by 12% and 37% compared to B-MAC+ and XY-MAC, respectively. It is possible to improve energy efficiency by operating under B-MAC+ for communication traffic on downlink and by operating under XY-MAC for communication traffic on uplink.

Finally, the data collection rate of ELA-MAC and XY-MAC, which apply early ACK for communication traffic on uplink, is higher than B-MAC+. In addition, the data collection rate of B-MAC+, which does not apply early ACK for communication traffic on uplink, is lower as the length of sleep duration increases.

In future studies, it is possible to consider the back-off approach in the preamble sampling technique. Back-off time must be adjusted properly because it influences both communication time and energy consumption. With the preamble sampling technique, the node occupying the channel continuously transmits several preamble packets during a certain time period. The preamble packets must be transmitted during a fixed time period similar to sleep duration without using an early ACK. On the other hand, the preamble packets could be transmitted during a time period shorter than sleep duration with using an early ACK. A time period that is half the sleep duration was applied to the backoff time in this experiment. However, we expect that more meaningful results can be obtained by adjusting the backoff time according to the presence or absence of an early ACK.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Industrial Strategic Technology Development Program, no. 10041947, funded by the Ministry of Science, ICT and Future Planning (MSIP, Korea). This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2014-H0301-14-1048) supervised by the NIPA (National IT Industry Promotion Agency).

References

[1] I. Demirkol, C. Ersoy, and F. Alagöz, “MAC protocols for wireless sensor networks: a survey,” IEEE Communications Magazine, vol. 44, no. 4, pp. 115–121, 2006.

[2] G. Anastasia, M. Contib, M. Francescoa, and A. Passarellab, “Energy conservation in wireless sensor networks: a survey,” Elsevier Ad Hoc Networks, vol. 7, no. 3, pp. 537–568, 2009.

[3] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” in Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ‘02), vol. 3, pp. 1567–1576, 2002.

[4] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” in Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys ’04), pp. 95–107, November 2004.

[5] M. Avvenuti, P. Corsini, P. Masi, and A. Vecchio, “Increasing the efficiency of preamble sampling protocols for wireless sensor networks,” in Proceedings of the 1st Mobile Computing and Wireless Communications International Conference (MCWC ’06), pp. 117–122, September 2006.

[6] M. Buettner, G. V. Yee, E. Anderson, and R. Han, “X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks,” in Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys ’06), pp. 307–320, November 2006.

[7] Z. Lu, T. Luo, and X. Wang, “XY-MAC: a short preamble MAC with sharpened pauses for wireless sensor networks,” in Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’12), pp. 1550–1554, Shanghai, China, April 2012.

[8] G. Lu, B. Krishnamachari, and C. S. Raghavendra, “An adaptive energy efficient and low-latency MAC for data gathering in wireless sensor networks,” in Proceedings of 18th International Parallel and Distributed Processing Symposium, vol. 30, pp. 26–30, 2004.

[9] Texas Instruments, DNS505. RSSI Interpretation and Timing.
[10] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[11] M. Liu, J. Cao, G. Chen, and X. Wang, "An energy-aware routing protocol in wireless sensor networks," *Sensors*, vol. 9, no. 1, pp. 445–462, 2009.

[12] A. Mollanejad, L. M. Khanli, M. Zeynali, H. Bahrbegi, and A. A. Alasti, "EHRP: novel energy-aware hierarchical routing protocol in wireless sensor network," in *Proceedings of the International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT '10)*, pp. 970–975, October 2010.

[13] CC430F6137, Texas Instruments, 16-Bit Ultra-Low-Power MCU, 32KB Flash, 4KB RAM, CC1101 Radio, CC430, http://www.ti.com/product/cc430f6137.
Submit your manuscripts at
http://www.hindawi.com