An event-aware MAC scheduling for energy efficient aggregation in wireless sensor networks

Donggook Kim, Jaesub Kim, Kyu Ho Park

Computer Engineering Research Laboratory, Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea

Abstract

The data aggregation scheme of wireless sensor networks (WSNs) can reduce the number of transmitted packets by aggregating multiple packets in one packet; thus, it can reduce energy consumption at the same time. However, the energy consumption of idle listening in WSNs remains dominant in the total energy consumption for WSNs. Therefore, the duty cycle MAC protocols have been used to reduce idle listening. However, if aggregation is used on duty cycle MAC protocols, it has very low performance caused by the latency of the sleep-wake scheduling.

The goal of the present work is to design energy efficient event-aware MAC scheduling that can be used by a data aggregation technique. In order to aggregate the packets spatially and temporally, two corresponding mechanisms are proposed: the event-aware and energy-aware routing (EE routing) protocol at the routing layer, and the aggregation scheduling MAC (A-MAC) protocol at the MAC layer. The EE routing protocol chooses the best path for better aggregation and for energy balance in WSNs. The A-MAC protocol adjusts schedules to aggregate more packets while maintaining a low latency.

The performances of the proposed protocols are evaluated through ns2 simulations. The results indicate that the proposed protocols reduce energy consumption by 80–90% and have balanced energy consumption while maintaining similar latencies and aggregation rates compared with previous aggregation protocols.

1. Introduction

Wireless sensor networks (WSNs) are used widely for sensing various events, such as forest fires, intruders, animal movements, gas leaks, and so on. These events may occur infrequently; thus, the sensor nodes should use little energy as possible when in an idle state, because most sensor nodes operate with batteries, which are often difficult to replace or recharge. Once an event occurs, however, multiple nodes near the event send data simultaneously, leading to competition between and collisions of data packets. Gupta and Kumar [1] showed that the throughput per node is \( \Omega \left( \frac{1}{\sqrt{n}} \right) \), even under optimal circumstances for \( n \) nodes. As a result, the bursty traffic causes a long latency and high energy consumption. However, Pottie and Kaiser [2] reported that the energy consumption for executing 3000 instructions is equivalent to sending 1 bit 100 m by radio; thus, it would be better to reduce the packet size using the CPU resources. Because nodes near the event have dependent data, the number of data packets decreases with the data aggregation scheme, thus reducing the latency and energy consumption of WSNs.

Data aggregation requires cooperation between the application layer and the network layer. The application layer aggregates data packets by sharing data packet headers, compressing data, processing data according to...
functions (min, max, average, count, event position, etc.), or removing duplications. The network layer assists in the collection of nearby packets and waits for the arrival of additional packets using spatial locality and temporal locality, respectively. The focus of this paper is the network layer that helps the data aggregation of the application layer. Data can be collected at a node with the assistance of the network layer, and the collected data can be aggregated using the data aggregation function of the application layer. The existing data aggregation protocols in the network layer can be classified [3] into two approaches: structured approaches and structure-free approaches.

Structured approaches [4–12] are suitable for static data gathering applications where fixed nodes send data periodically, because packets can be easily aggregated when nodes send packets to parent nodes along pre-constructed structures and the parent nodes await the packets of the child nodes for a fixed time. However, in dynamic event triggered applications where the traffic patterns are time varying, structure-based approaches are not efficient. Because a parent node does not know whether its child nodes have data, it cannot determine the length of the wait time. A short wait time leads to fewer aggregations and a long wait time leads to a higher latency. Furthermore, packets are not aggregated well because the traffic patterns change but the structure is based on the previous traffic pattern. Therefore, some structured approaches [4,5] reconstruct the structure for current event distributions, but the overhead of the structure reconstruction and maintenance could be larger than the benefit of data aggregation.

To reduce the overhead of structure maintenance, structure-free approaches have recently been proposed for event triggered applications [13,3,14]. One structure-free approach [3] uses a data-aware anycast and random wait approach, hereafter called the DR protocol. To increase the aggregation rate, a node attempts to route its packet to another node that has a packet. Because the sender node does not know whether neighbor nodes have packets, the DR protocol uses anycast, which does not specify a destination when sending packets. Multiple receiver nodes compete with others, and the ‘winning’ node is chosen as a receiver node. To collect additional packets, nodes using the DR protocol wait for a random amount of time before beginning transmission. However, all sensor nodes in the DR protocol must operate on an always awake network to aggregate or forward data.

Wireless interfaces of sensor nodes consume large amounts of energy when they are awake although they are not active, which is called idle listening. Table 1 shows the power consumption of various radio states in CC1000 [17] and CC2420 [19] RF transceivers, which are widely used in sensor devices. Although the transmitting power has the largest consumption, the idle listening power is also large and is similar to the receiving power.

Table 2 shows the energy consumption of a sender node with and without data aggregation in a simple environment. In this test, the sender transmits 500 packets with no aggregation and 50 packets with 90% aggregation over 100 s. The energy consumption of the data transmission decreases when aggregation is used, but the energy consumption of idle listening increases because the idle listening time increases with the reduced number of data transmissions. As a result, the total energy consumption does not significantly decrease with aggregation. In WSNs, the energy consumption during idle listening is considerable. Therefore, to achieve energy efficiency in WSNs, idle listening must be reduced in addition to increasing the aggregation rate.

Several previous works on reducing idle listening have been reported. The sensor MAC (S-MAC) [20,21] protocol reduces idle listening by using periodic listen and sleep cycles, though it also reduces the bandwidth and causes delays at each hop. Thus, the S-MAC protocol has a high latency when there are many packets to send or when the hop-count to the destination increases.

To reduce the latency in multi-hop forwarding, several multi-hop scheduling protocols such as the look-ahead schedule MAC (LAS-MAC) [22,23] protocol and the routing-enhanced MAC (R-MAC) [24] protocol have been proposed, which reserve packet forwarding schedules across multiple nodes in the listen period and forward data in the sleep period by following the reserved schedules.

**Table 1**

Power consumption of the Mica2 radio (CC1000) and the TelosB radio (CC2420) [15].

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Communication chip</th>
<th>Power consumption, Transmit (mW)</th>
<th>Receive (mW)</th>
<th>Listen (mW)</th>
<th>Sleep (µW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mica2 [16]</td>
<td>CC1000 [17]</td>
<td>31.2</td>
<td>22.2</td>
<td>22.2</td>
<td>3</td>
</tr>
<tr>
<td>TelosB [18]</td>
<td>CC2420 [19]</td>
<td>52.2</td>
<td>56.4</td>
<td>56.4</td>
<td>3</td>
</tr>
</tbody>
</table>

**Table 2**

Comparison of the energy consumption without data aggregation (500 packets transmission during 100 s) and with data aggregation (50 packets transmission during 100 s). TX: transmit, IDLE: idle listen.

<table>
<thead>
<tr>
<th>H/W</th>
<th>Energy w/o agg. (J)</th>
<th>Energy w/agg. (J)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TX</td>
<td>IDLE</td>
</tr>
<tr>
<td>Mica2</td>
<td>1.1</td>
<td>1.4</td>
</tr>
<tr>
<td>TelosB</td>
<td>2.0</td>
<td>3.5</td>
</tr>
</tbody>
</table>
However, these protocols do not consider data aggregation nor do they construct an appropriate path for data aggregation. These protocols reserve multiple data schedules independently and send each data piece as soon as possible; hence, data aggregation does not occur well. Therefore, both data aggregation and the idle listening problem must be considered in order to reduce the total energy consumption.

To solve the above challenges, an event-aware and energy-aware routing (EE routing) protocol and an aggregation-MAC (A-MAC) protocol is proposed and referred to as the EA protocol hereafter. The goal in this study is to design energy efficient event-aware MAC scheduling that could be used by a data aggregation technique. The EE routing protocol selects the path for better aggregation and energy balance in WSNs. The A-MAC protocol allows forwarding schedules across multiple nodes with the help of the EE routing protocol in the listen period and sends data according to the reserved schedules in the sleep period. In the listen period, the nodes that overhear the schedules of neighbors opportunistically join the schedules of neighbors for aggregation (schedule join mechanism). The A-MAC protocol creates space between schedules for the potential joining of neighbors (schedule margin mechanism) and rearranges the schedules to increase the aggregation rate (schedule shift mechanism).

This work contributes the following to the aggregation protocol in WSNs.

1. An analysis of the importance of idle listening in the aggregation protocol.
2. A proposal for methods of generating an aggregation structure when events occur.
3. A proposal for new multi-hop scheduling approaches to increase the aggregation rate by adjusting the transmission order.
4. Validation of the operation by means of an extensive simulation.

The remainder of this paper is structured as follows. We describe the background and related work in Section 2. Section 3 presents the EE routing protocol and the A-MAC protocol. In Section 4, we compare the performance of the EA protocol to that of the DR protocol via a simulation. Finally, Section 5 concludes the paper.

2. Related work

2.1. Structure-based approaches

Fig. 1 shows the aggregation procedure of a structure-based approach. Structured approaches [4–12] construct a structure that connects all sensor nodes, and fixed nodes periodically send data to a sink node along the structure in static data gathering applications. For event triggered applications where the traffic pattern changes continuously, structured approaches are not appropriate because structure-based approaches have a low aggregation rate due to the data only being sent along pre-constructed structure, a high latency due to waiting for all data from the child nodes, and poor energy balance as a result of using a fixed structure in event triggered applications. The overhead of structure reconstruction and maintenance could be larger than the benefit of data aggregation if the structure is reconstructed to make a new structure for the current event distribution.

2.2. Structure-free approaches

In an effort to reduce the overhead of the structure construction and maintenance, structure-free approaches have been introduced [13,3,14]. Because there is no structure, nodes make two decisions before sending data locally: where to send the data and when to send the data. The DR protocol [3] uses data-aware anycast and random wait: Fig. 2 shows the aggregation procedure of the DR protocol. Because the sender node does not know whether the neighbor node has a packet, the DR protocol uses anycast. The sender node sends a Request-To-Send (RTS) packet with no destination; a receiver node sends a Clear-To-Send (CTS) packet. The receiver nodes wait according to their priority before sending the CTS packet. Each node has a different priority class according to its position and whether the node has data. Class A nodes are nodes that are closer to the sink node than the sender node and that have data to aggregate. Class B nodes are nodes that have data but which are further from the sink node. Class C nodes are nodes that have no data but are closer to the sink node. Class A nodes send a CTS packet earlier than class B nodes, and class B nodes send a CTS packet earlier than class C nodes. When a node hears the CTS packet from other nodes, it cancels its CTS transmission. To avoid collisions
with other nodes in the same class, nodes in the same class transmit CTS packets after a random delay during the duration of the class.

However, to aggregate or to forward data, the DR protocol must be operated in an always awake network because receiver nodes should send CTS packets whenever sender nodes send RTS packets. Thus, nodes in a DR protocol system consume large amounts of energy during idle listening. Even though the DR protocol reduces the energy consumption of the data transmission via data aggregation, the total energy reduction is not significant because the energy consumption of idle listening remains dominant.

The I-FrameComm protocol [14] performs aggregation on the B-MAC protocol [25]. The listen times of sensor nodes are not synchronized, so a sender transmits a trail of identical packets for the receiver to catch during its listen time. When an immediate neighbor node with a packet to aggregate overhears the packet transmission of the sender, it sends an interrupt packet to the sender and sends data to the receiver for aggregation. The I-FrameComm protocol also aggregates data opportunistically, but it has a high latency in a multi-hop network because the sender must wait for the receiver nodes to wake up at each hop.

2.3. Approaches for reducing idle listening

Because the power consumption of idle listening is much higher than that of the sleeping state, several MAC protocols have been developed to reduce idle listening. The sensor MAC (S-MAC) [20,21] protocol reduces idle listening using periodic listen and sleep cycles. Nodes exchange synchronization packets, so they wake up and sleep at the same time in a time-synchronized manner. The S-MAC protocol sends control packets and data packets only during the listen period and sleeps during the sleep period. When the listen period is over, the node can send data in the next listen period. Thus, the S-MAC protocol has a high latency because it should wait for the next listen period when there are many packets to send or when the hop-count to the destination is high. Therefore, if the DR protocol is operated on the sleep-wake schedule as with the S-MAC protocol, it will have low energy consumption but a high latency.

Multi-hop scheduling protocols, such as the look-ahead schedule MAC (LAS-MAC) protocol [22,23] and the routing-enhanced MAC (R-MAC) protocol, are also based on periodic listen and sleep cycles as in the S-MAC protocol, but they have lower latency than the S-MAC protocol because they send control packets more hops during the listen period.

The proposed protocol follows the approach of the multi-hop scheduling protocols, which use a periodic listen and sleep strategies to reduce energy consumption, and reserve schedules in the listen period to decrease latency. However, they are not designed for aggregation. Fig. 3 shows the aggregation procedure of the LAS-MAC protocol. The LAS-MAC protocol selects a next hop without consideration of the event distribution. The LAS-MAC protocol does not wait to receive other data, but rather makes schedules to send the packet as soon as possible.

2.4. Summary

Table 3 shows a comparison of the EA protocol with previous aggregation protocols. Most of previous works are query-based and have a static structure. The DR protocol is an event-based structure-free protocol, but it cannot operate on the sleep-wake scheduling.

The EA protocol is the first event-based structure-free protocol that is operated on the sleep-wake scheduling. The event-based structure-free protocol that is operated on the sleep-wake scheduling cannot be achieved by simply combining the multi-hop scheduling protocol and the DR protocol. The DR protocol has the time overhead for waiting for the CTS packet and for waiting for the packet of neighboring nodes, so it consumes a great deal of time to transmit a control packet at each hop. Thus, the DR protocol cannot be operated well with the multi-hop scheduling protocol and has high latency. Moreover, the multi-hop scheduling protocols are not appropriate for aggregation because they only make schedules for the packet forwarding.

Therefore, we designed the EA protocol to utilize the advantage of the multi-hop scheduling protocol. The EA protocol selects a routing path in the direction that events may occur and reserves schedules for aggregating the data packets of neighboring nodes by using event prediction. Neighboring nodes can join the schedules after hearing
the neighboring schedules. Furthermore, the schedules can be modified dynamically when the event prediction is wrong.

### 3. Event-aware and energy-aware routing protocol and aggregation-MAC protocol

#### 3.1. Overview

In this paper, an EA protocol is proposed: it is a hybrid of the event-aware and energy-aware routing (EE routing) protocol and the aggregation-MAC (A-MAC) protocol. There are two decisions for data aggregation:

- **Where to send the data for spatial locality**
  - The EE routing protocol: Nodes decide the next hop that is likely to have data to be aggregated and retains a large amount of energy.

- **When to send the data for temporal locality**
  - Schedule margin of the A-MAC protocol: When a node makes a schedule, it creates an empty space inside the schedule. The node predicts the number of joins of neighbor nodes and uses it for the size of the empty space.
  - Schedule join of the A-MAC protocol: When the node overhears schedules from neighbor nodes, it can join an empty space of ongoing schedules. This mechanism is called schedule join.
  - Schedule shift of the A-MAC protocol: When there is insufficient space to accept the join request of neighbor nodes with incorrect predictions, the node shifts the schedules to accept the join request of neighbor nodes.

In the listen period, the A-MAC protocol reserves schedules and the multi-hop path is determined by the EE routing protocol. When the nodes make schedules, they set empty space, called a schedule margin, inside the schedules for the possible joins of neighbors. Neighbor nodes that overhear the control packets try to join those schedules. The A-MAC protocol rearranges the schedules using the schedule shift to increase the aggregation rate if needed. In the sleep period, only those nodes involved wake up, send, receive, and aggregate the data packets according to the schedule. The detailed operation will be explained in the following sections.

#### 3.2. Event-aware and energy-aware routing (EE routing) protocol

The event-aware and energy-aware routing (EE routing) protocol is used to select paths for aggregation and for energy balance. It is assumed that the nodes know their coordinates from a global positioning system (GPS) or other localization protocols [26,27]. There are two primary purposes of the EE routing protocol: to increase the aggregation rate and to balance the energy consumption distribution. To increase the probability of data aggregation, the EE routing protocol chooses the next hop that is most likely to have data. In order to find the nodes that are likely to have data to send, a node traces the event rate information of its neighbor nodes. The event rate is the average number of event occurrences in the unit time. In order to balance the energy consumption distribution, the EE routing protocol chooses the next hop that has a larger energy because a node that aggregates the data of its neighbor nodes consumes more energy to gather and aggregate...
the data packets. To achieve these purposes, the EE routing protocol operates as follows:

1. Using a neighbor table, a node finds candidate nodes that are closer to the sink node than itself.
2. Among the closer nodes, it selects candidate nodes that have a higher event rate than the average event rate of the candidate nodes.
3. Among the nodes with above average event rates, the node selects the next hop nodes that have higher remaining energies than the average remaining energy of the candidate nodes.
4. If there are two or more candidate nodes, the node selects the node nearest to the sink node.

With the EE routing protocol, a node selects the next hop that has higher probability to have a packet to send and a higher remaining energy. In order to trace the event rate of its neighbor nodes, the node sends a control packet with a flag when it has a packet to send. The other nodes check the control packet during the listen period and accumulate the number of event occurrences in their neighbor tables. The event occurrence value of the neighbor table is updated using the exponentially weighted moving average (EWMA). Then, the event rate of a sensor node \( i \) (\( ER_i \)) is represented as:

\[
ER_i = ER_i \times (1 - \alpha) + ER_{current_i} \times \alpha,
\]

where \( ER_{current_i} \) is the observed event rate in the current listen period. A weighting factor, \( \alpha \), is chosen by the user according to the event characteristics such as the event speed. The remaining energies of the neighbor nodes are acquired from the synchronization packets of neighbors with additional remaining energy information. Packets are forwarded to a sink node when there are nodes closer to the sink node than the sender node. When there are holes in a network and the sender does not have a node that is closer to the sink node, it avoids the holes using the perimeter mode forwarding from the Greedy Perimeter Stateless Routing (GPSR) protocol [28].

3.3. Schedule setup of the aggregation-MAC (A-MAC) protocol

Each node maintains its schedule table, which maintains the reserved operations for the sleep period. For simplicity, it is assumed that applications process data according to functions such as \( MIN \), \( MAX \), \( AVERAGE \), \( COUNT \), and \( EVENT \) \( POSITION \), in which many packets can be aggregated into one packet. The sleep period is divided into identically sized schedule slots. The schedule slot is used instead of the schedule time in order to decrease the overhead of the control packet by reducing the packet size. The node forwards the data packet after a Short Inter-Frame Space (SIFS), which is a sufficiently long time to process the packet, to aggregate the packet, and to switch radio modes (from a receiving state to a transmitting state). The schedule slot size is the sum of the data transmission time, acknowledgment transmission time, and SIFS after each process.

There are three types of schedules: \( S \) (send), \( R \) (receive), and \( N \) (network allocation vector). In the \( S \) schedule, the node wakes up, sends a packet, receives an acknowledgment, and sleeps. In the \( R \) schedule, the node wakes up, receives a packet, sends an acknowledgment, and sleeps. In the \( N \) schedule, the node sleeps to avoid collisions with the neighboring \( S \) and \( R \) schedules.

The schedules are established by forwarding an aggregation RTS (A-RTS) packet. Because the wireless transmission has broadcast characteristics, the A-RTS packet of nodes in a multi-hop transmission can be simultaneously used as a CTS packet and an RTS packet in order to reduce the redundant header fields of the CTS and RTS packets. The A-RTS packet contains a final destination address, a schedule slot number, a previous node address, and a previous schedule slot number in addition to the original RTS packet. The final destination address is used to find the next forward hop using the EE routing protocol. Because the MAC protocols do not know the next hop node for the final destination address, the EE routing protocol receives the final destination address from the MAC layer and returns the next forward hop to the MAC layer when the node receives the A-RTS packet. The schedule slot number is used to reserve the new \( S \) schedule of the current node, to reserve the new \( R \) schedule of the target node, and to make the \( N \) schedules of the sender node’s neighboring nodes. The previous node address and the previous schedule slot number are used to confirm the \( S \) schedule of the previous hop node, to confirm the \( R \) schedule of the current node, and to make the \( N \) schedules of the receiver node’s neighboring nodes.

Fig. 4 shows the process of the schedule setup in the listen period and the data transmission in the sleep period. The node schedule tables according to the process are depicted in Table 4. This process completes the following: in the listen period, nodes exchange the A-RTS packets and
reserve schedules in their schedule tables, as shown in procedures from (A) to (C) in Fig. 4 and Table 4.

(A) Node A sends the A-RTS packet. Node B receives the A-RTS packet from node A.
1. Node A finds that the next hop is node B with the final destination node E from the EE routing protocol. Node A finds that Slot 1 is empty in its schedule table and reserves an estimated S schedule in Slot 1. It sends an A-RTS packet, which contains the information: the sender node is node A, the target node is node B, the final destination node is node E, and the schedule slot number is 1.
2. When node B receives the A-RTS packet from node A, it reserves an estimated R schedule in Slot 1 if there are no prior schedules in Slot 1. The estimated schedule is kept until node B sends the A-RTS packet as a confirmation packet.

(B) Node B sends the A-RTS packet. Node A and node C receive the A-RTS packet from node B.
1. Node B finds that the next hop is node C with the final destination node E from the EE routing protocol. Node B finds that Slot 2 is empty in its schedule table and reserves an estimated S schedule in Slot 2. It sends the A-RTS packet with the information: the sender node is node B, the target node is node C, the final destination node is node E, the schedule slot number is 2, the previous node address is node A, and the previous schedule slot number is 1. Node B changes the estimated R schedule to the confirmed R schedule in Slot 1 in its schedule table if the A-RTS packet is transmitted without a problem.
2. When node A receives the A-RTS packet from node B, it changes the estimated S schedule to the confirmed S schedule in Slot 1, because the both nodes agree on this schedule. Node A makes the N schedule in Slot 2 in its schedule table, in order to avoid collisions during node B’s transmission.
3. When node C receives the A-RTS packet from node B, it reserves an estimated R schedule in Slot 2 and makes the N schedule in Slot 1 in its schedule table, in order to avoid collisions during the reception of the data from node B.

(C) Node C sends the A-RTS packet. Node B receives the A-RTS packet from node C.

1. Node C sends the A-RTS packet with the information: the sender node is node C, the previous node address is node B, and the previous schedule slot number is 2. Node C changes the estimated R schedule to the confirmed R schedule in Slot 2 in its schedule table.
2. When node B receives the A-RTS packet from node C, it changes the estimated S schedule to the confirmed S schedule in Slot 2 in its schedule table.

This procedure continues until the listen period ends or the A-RTS packet reaches its destination. In the sleep period, all nodes sleep and behave according to their reserved schedules, as shown in procedures from (D) to (G) in Fig. 4.

(D) Node A sends a data packet to node B at the time of Slot 1.

(E) Node B sends an acknowledgment packet to node A. Node A then sleeps.

(F) Node B sends a data packet to node C at the time of Slot 2.

(G) Node C sends an acknowledgment packet to node B. Node B and node C then go to sleep.

After the nodes operate according to the reserved schedules, they go to sleep until the next listen period.

3.4. Schedule margin

The A-MAC protocol makes space, called schedule margins, between schedules to accept the possible joins of neighbor nodes with minimal schedule modification. As shown in Fig. 5, node B makes its S schedule in Slot 3 after the schedule margin of Slot 2 from the R schedule in Slot 1. The schedule margin is the duration during which the predicted nodes send data. By using the collected event rate information again, which is used in the EE routing protocol,
the node can predict the number of neighbors that may have data to send. From the event rate information of the neighbor table of node B, node B predicts that there is one node, node G, which may join node B. Therefore, node B sets a schedule margin of one slot.

The expected number of neighbors that will send data is calculated from the number of neighbor nodes that have an event rate over the threshold. In the proposed protocol, the threshold value is the average event rate of all neighbor nodes. The administrator can adjust the threshold value according to the characteristics of the application. The smaller the threshold is, the higher the schedule margin is; thus, the protocol will have a high aggregation rate, but will also have a high average latency. The larger the threshold value is, the smaller the schedule margin is, and the aggregation rate will be lower. However, a larger threshold value is preferred because the position of the schedule can be adjusted with the schedule shift mechanism, which will be explained in Section 3.6.

3.5. Schedule join

When node B exchanges the A-RTS packet to make schedules, node G will overhear the A-RTS packet of node B and will know the schedules of the neighboring node B. As can be seen in Fig. 6, node G can join the empty slot created by the schedule margin of node B with an A-RTS packet with the information that the sender node is node G, the target node is node B, and the schedule slot number is 2. The A-RTS packet does not need to be forwarded to the final destination, node E, further than node C, because the packet of node G will be aggregated with that of node B, and node B has already reserved the schedule for the final destination, node E.

3.6. Schedule shift

When there are more schedule joins than expected, a node loses the chance to aggregate more data packets. Thus the schedule shift mechanism that shifts the schedule in order to aggregate more packets is suggested for implementation in this process. In Fig. 7(a), node B makes the S schedule for node C at Slot 2 with the assumption that there is no join of neighboring nodes. In Fig. 7(b), node B receives the schedule join of node G and reserves the R schedule from node G in Slot 3. If node B does not modify its schedules, node B will receive a data packet from node A, send it to node C, and then receive another data packet.

![Fig. 6. Schedule join.](image9.png)

![Fig. 7. An example of schedule shift.](image10.png)
from node G later; thus, it cannot aggregate the two data packets.

Therefore, if node B can send an additional A-RTS packet that updates its schedule in the listen period, it sends an A-RTS packet that is used to shift the schedule in order to send the data packets after aggregation of the two packets. Node B sends an A-RTS packet with the information that the sender node is node B, the target node is node C, and the schedule slot number is 4. When node C receives the A-RTS packet from node B, it already has the R schedule from node B at Slot 2. So, node C cancels the previous R schedule for Slot 2 and makes a new R schedule in Slot 4. As shown in Fig. 7(c), with the schedule shift mechanism, node B moves the S schedule from Slot 2 to Slot 4, and node C moves the R schedule from Slot 2 to Slot 4, thus sending the data packet after receiving all other data packets during the sleeping time. Therefore, although the number of schedule joins exceeds the expected number, the schedule shift mechanism adjusts the previous schedules with a low overhead, so the A-MAC protocol achieves a higher aggregation rate.

3.7. Summary

Fig. 8 shows the data aggregation procedure of the EA protocol. As shown in Fig. 8(a), the EA protocol reserves the schedule with control packets along the multi-hop path with the assistance of the EE routing protocol. The path is selected for aggregating more packets and for energy balance. The nodes make a schedule with a schedule margin, so neighbor nodes can join to the schedule. Node D overhears the A-RTS packet of node E and joins the schedules, as shown in Fig. 8(b). In the sleep period, the nodes send data packets according to the reserved aggregation schedule as shown in Fig. 8(c). As shown in Fig. 8(d), data aggregation occurs in node E with data transmission according to the schedules.

3.8. Discussion

The event rate is updated when the neighboring sensor nodes that detect the event send the A-RTS packet via piggybacking whether the event occurs or not. Thus, the event rate is not increased for the intermediate nodes, which do not detect the event but only forward the A-RTS packet. The event rate is updated only when the event is detected without regard for the duplicate-sensitive aggregation or the duplicate-insensitive aggregation. This is because a node can select as the next hop node any among the nodes that have duplicate data packets. Whether the data packet is duplicated or not is judged not when the routing path and the schedule are made using the A-RTS packets, but when the actual data packets are transmitted and collected. Therefore, the A-RTS packet is transmitted to the direction in which more events can be collected by using only the event rate. The goal of the transmission of A-RTS packets is to collect the events that will be transferred to the same sink.

4. Performance evaluation

4.1. Simulation setup

The performance of the EA protocol was simulated using the ns2 network simulator (version 2.33) [29] and was compared with that of the DR protocol, because the DR protocol has a lower latency and higher aggregation ratio than the structured approach [3]. In this simulation, all protocols aggregate every packet of events. The network is a 900 m by 900 m square region with a grid topology, which consists of 49 nodes separated by 140 m. The sink node is located at one corner of the grid network. The center of the event moves according to the random way point model. Unless otherwise stated, the event moving speed is 30 m/s for 1900 s with a pause time of 0 s, the radius of the event is 200 m; and the event generation interval is 10 s. If nodes are inside the event radius, they generate data packets with a 50 bytes payload periodically according to the event interval and send data packets to the sink node. The duty cycle is set to 10%; the listen period is 591 ms; the synchronization period is 55.2 ms; and the sleep period is 5.231 s. A weighting factor, \( \alpha \), of 0.5 is used because the location where the event occurs continuously changes.

The simulation included a sensor node with the Two Ray Ground radio propagation model in the air, a single omni-directional antenna, and 20 kbps wireless interface
with a 250 m transmission range and a 550 m carrier sensing range, which works in a similar manner to the 914 MHz Lucent WaveLAN direct sequence spread spectrum (DSSS) radio interface. To evaluate the energy consumption, the energy model used had $P_{\text{transmission}} = 2.4$ mW, $P_{\text{receive}} = 13$ mW, and $P_{\text{idle}} = 13$ mW, where $P$ represents power; these parameters are sourced from Model TR1000 of RF Monolithics Inc. [30].

The sizes and transmission times of packets used in the simulations are presented in Table 5. The nodes sense carriers during a Distributed Inter-Frame Space (DIFS) and a random time lower than the contention window (CW) as in the IEEE 802.11 protocol. In the simulation, DIFS is 10 ms, SIFS is 5 ms and the CW is 64 ms. The schedule slot time is the duration of the data packet and the acknowledgment packet including the SIFS after each packet, which is 64 ms.

To compare these protocols, the three metrics listed below are used:

- **Average latency:** The average delay of all received packets at the sink.
- **Aggregation rate:** The ratio of aggregated packets to total received packets at the sink. The aggregation rate will be higher if more packets are aggregated.
- **Power consumption:** The average power consumption of all nodes.

The performance of the DR protocol depends on the maximum random wait value. The larger the maximum random wait value is, the higher the average latency is. However, the smaller the maximum random wait value is, the lower the aggregation rate is. A maximum random wait value of 2 s was used because the DR protocol suggests 2 s as a default value for the random wait time [3].

### 4.2. Simulation results

In this section, the results for different event intervals are shown. The event interval varies from 2 s to 100 s. To reduce the total energy consumption, the DR protocol can be operated on a periodic listen and sleep scheme, hereafter called the DR-on-SMAC protocol. The EA protocol is also compared with the LAS-MAC protocol and the DR-on-SMAC protocol. The size of the LAS-RTS packet of the LAS-MAC protocol is 14 bytes. The LAS-MAC protocol uses the GPSR routing protocol [28] as its routing protocol.

#### 4.2.1. Average latency

In Fig. 9, the EA protocol has the lowest latency among the protocols compared. The DR protocol and the LAS-MAC protocol has two times higher latency than the EA protocol, and the DR-on-SMAC protocol has seven times higher latency than the EA protocol. For the DR protocol, the data-aware anycast mechanism takes an inefficient longer routing path to the sink node to aggregate more packets and the random wait mechanism increases the delay to aggregate possible packets. The DR-on-SMAC protocol has a much higher latency than the DR protocol due to the latency of the sleep-wake scheduling. The DR protocol and the DR-on-SMAC protocol attempt to aggregate the data packets of neighbor nodes even though the data packet is located in a backwards direction; thus, they have higher average latency as the event interval decreases because another data packet is generated at neighbor nodes before they send the data packet to the sink node. However, the EA protocol has a relatively low and constant average latency because the EA protocol reserves schedules to the sink node first, aggregates packets of neighboring nodes opportunistically in the listen time, and sends data according to schedules in the sleep time. The LAS-MAC protocol has a higher average latency than the EA protocol because more time is needed to transfer data packets with lower aggregation.

### Table 5

<table>
<thead>
<tr>
<th>Packet</th>
<th>Size (bytes)</th>
<th>Transmission time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTS/CTS/ACK/SYNC</td>
<td>10</td>
<td>11.0</td>
</tr>
<tr>
<td>A-RTS</td>
<td>16</td>
<td>15.8</td>
</tr>
<tr>
<td>DATA</td>
<td>50</td>
<td>43.0</td>
</tr>
</tbody>
</table>

Fig. 9. Average latency.

Fig. 10. Aggregation rate.
4.2.2. Aggregation rate

In Fig. 10, the LAS-MAC protocol has the lowest aggregation rate because it forwards data to the sink node without considering aggregation. As the event interval decreases, the number of packets to aggregate in the same node also increases, so the aggregation rate of the LAS-MAC protocol increases. The DR-on-SMAC protocol has the highest aggregation rate because the node does not forward data packets to the sink node, but rather forwards data packets to neighboring nodes that have data packets, even if it is located away from the sink node. Therefore, even though the DR-on-SMAC protocol has the highest aggregation rate, it has the longest average latency. The EA protocol has a similar aggregation rate to that of the DR protocol because the EA protocol chooses a path based on the event generation rate of neighboring nodes and reorders schedules.

4.2.3. Power consumption

The power consumption of the EA protocol, the LAS-MAC protocol, and the DR-on-SMAC protocol is eight times lower than that of the DR protocol, as depicted in Fig. 11. The DR protocol consumes a large amount of energy because it is always awake. The EA protocol, the LAS-MAC protocol, and the DR-on-SMAC protocol use a much lower energy, because they reduce idle listening by periodic listen and sleep cycles.

In summary, the EA protocol has a lower latency than the DR protocol and a similar aggregation rate to the DR protocol, while it consumes a much lower amount of energy than the DR protocol. Although the EA protocol has an overhead in obtaining the neighbor information, the overhead of the EA protocol is similar to that of the S-MAC protocol and the LAS-MAC protocol. Most of the obtained information is used to synchronize nodes to sleep and wake at the same time in a time-synchronized manner. The additional event count information occupies 1 bit in the packet size and about 1 byte in the memory of the sensor node. Another overhead is that the A-RTS packet size is larger than the control packet of other protocols because it includes more entries, such as the final destination address and the previous node address. However, by using the A-RTS packet for the RTS and CTS packets simultaneously, the overhead of the larger size control packet could be decreased. Furthermore, a schedule slot (1 byte) is used instead of a schedule time (2 bytes) by dividing the synchronized sleep time to equal sized slots. Therefore, the EA protocol including the overhead has a lower latency and lower energy consumption than other protocols.
4.2.4. Effect of event radius

In this simulation, the performance of the EA protocol, DR protocol and DR-on-SMAC protocol were evaluated by varying the event radius from 100 m to 900 m. The average latency with the varying event radius is depicted in Fig. 12(a). As the event radius enlarges, the number of packets in the network also increases. The DR-on-SMAC protocol collects the data packets in the network, although the movement direction is away from the sink node. Thus, the average latency of the DR-on-SMAC protocol increases as the number of packets in the network increases. The DR protocol does not have a sleeping time, thus it can collect

---

![Fig. 13. Results for different event moving speeds.](image)

---

![Fig. 14. The EA protocol with various routing protocols. EE routing: event and energy-aware routing, E routing: energy-aware routing, shortest: shortest path routing.](image)
data packets from neighboring nodes and send aggregated data packets to the sink node before other packets are generated. The EA protocol has a relatively constant average latency with the increasing event radius, because most packets are aggregated and arrive at the sink node simultaneously. From these results, it is concluded that the EA protocol operates well with a large event radius and maintains a low overhead.

For the aggregation rate, when the event radius is small, there are fewer nodes with packets; thus, the nodes are not able to aggregate the packets of neighboring nodes and maintain a low aggregation rate. The EA protocol has similar aggregation rate as the DR protocol and the DR-on-SMAC protocol, as shown in Fig. 12(b). Because the EA protocol aggregates the data packets of neighboring nodes at one node, it has a higher aggregation rate. As shown in Fig. 12(c), the energy consumption of the EA protocol and the DR-on-SMAC protocol are much lower than that of the DR protocol.

4.2.5. Effect of event speed

In this simulation, the performance of the EA protocol and the DR protocol were evaluated with varying event speeds from 0 m/s to 60 m/s. With a higher event speed, the prediction of the event position can be incorrect. However, it can be seen that the results remain steady at different speeds. As both protocols do not create a structure, mobility has little impact on the performance. In Fig. 13(a), the EA protocol has a lower average latency than the DR protocol. The aggregation rate of the EA protocol is similar to that of the DR protocol, as seen in Fig. 13(b). The energy consumption of the EA protocol is much lower than that of the DR protocol, as illustrated in Fig. 13(c).

4.2.6. Effect of the EE routing protocol

In this simulation, the performance of the EA protocol is evaluated while varying the routing protocols. The event speed is 0 m/s to emphasize the effect of energy balance. The shortest path routing protocol, the energy-aware routing protocol, and the EE routing protocol are compared.

Fig. 14(a) shows the average latency with different event intervals. The EA protocol with the EE routing protocol has the lowest average latency because it chooses the path to aggregate more packets and the average latency decreases with data aggregation.

![Fig. 15. Energy consumption distribution of the EA protocol with various routing protocols and of the DR protocol.](image_url)

(a) The shortest path routing protocol.  
(b) The Event-aware and Energy-aware routing (EE-routing) protocol.  
(c) The energy-aware routing protocol.  
(d) The Data-aware anycast and Random-wait (DR) protocol.

![Fig. 16. Node distribution of a realistic scenario.](image_url)
Fig. 14(b) shows the aggregation rate with different event intervals. The EA protocol with the EE routing protocol has the highest aggregation rate. Thus, the EA protocol with the EE routing protocol selects a more efficient path for aggregation.

In Fig. 14(c), the shortest path routing has the lowest energy consumption because it chooses the shortest path; thus, minimal nodes are used for forwarding the data packets. However, the shortest path routing has an unbalanced energy consumption, which is shown in next section. The energy-aware routing (E routing) has the largest energy consumption because it chooses the next hop node that has the largest energy; thus, the maximum number of nodes is used. EE routing has a smaller energy consumption than E routing because it chooses the next hop that has a packet to aggregate, thus reducing the number of packet transmissions by aggregating the data packets.

4.2.7. Energy consumption distribution of the ee routing protocol

The energy consumption distributions of the EA protocol with varying routing protocols are compared. Fig. 15 shows the event distribution and the consumed energy of each node in the topology. The event interval is 40 s. The shortest path routing protocol consumes more energy in the static path, as depicted in Fig. 15(a). The EE routing protocol balances the energy consumption distribution across nodes and has the lowest energy consumption, as can be seen in Fig. 15(b). The energy-aware routing protocol also balances the energy consumption distribution, but it does not select the path in which the node aggregates the packets of neighbor nodes. Therefore, the energy-aware routing protocol causes more traffic with a lower aggregation rate. Thus, the energy-aware routing protocol has a higher average energy consumption than the EE routing protocol, and it consumes more energy near the sink nodes with a higher number of packets, as depicted in Fig. 15(c).

As shown in Fig. 15(d), the DR protocol consumed the most energy with idle listening; thus, every node has a high energy consumption. Hence, by using the A-MAC protocol, the EA protocol reduces the energy consumption of idle listening. Also, by selecting an efficient path for aggregation and energy balance, the EA protocol with the EE-routing protocol has the lowest and most balanced energy consumption.

4.2.8. Realistic scenario

In this simulation the performance of the EA protocol and the DR protocol are evaluated in a realistic scenario. Fig. 16 shows the node distribution of a realistic scenario. The total number of nodes is 120 and they are uniform randomly distributed in 900 m by 900 m square area. The event radius is 200 m with varying speeds from 3 m/s to 15 m/s.

Fig. 17(a) shows that the average latency of the EA protocol is lower than the DR protocol in the realistic scenario. In Fig. 17(b), the EA protocol has a higher aggregation rate than the DR protocol. The power consumption of the EA protocol is much lower than the DR protocol. In the realistic scenario presented, the EA protocol maintains a lower latency and higher aggregation rate than the DR protocol while having a lower power consumption.

The EA protocol consumes 10–20% of the energy consumption of the DR protocol, but the EA protocol has equivalent or better performance (latency, aggregation rate) than the DR protocol. The EA protocol reduces the idle
listening time by using the sleep-wake scheduling and uses smart scheduling mechanisms (EE-routing, schedule margin, schedule join, schedule shift) in order to collect many packets and aggregate them during a short listening period; thus, the EA protocol has similar performance to the DR protocol although the EA protocol consumes much less energy. If only the sleep-wake scheduling is used, the energy consumption will be decreased, but the latency will increase and the aggregation rate will decrease.

5. Conclusion

In this paper, an EA protocol, which consists of the event-aware and energy-aware routing (EE routing) protocol and the aggregation MAC (A-MAC) protocol, was proposed. It achieves energy efficiencies by reducing idle listening with periodic listen and sleep cycles. In order to aggregate more packets and balance the energy consumption, the EE routing protocol was suggested. In order to reduce the increase of latency due to the sleep-wake scheduling, the A-MAC protocol uses the listening time for multi-hop scheduling for aggregation. In the listen period, the nodes that overhear the schedule of neighboring nodes opportunistically join the schedules of the neighboring nodes for aggregation. The schedule margin and schedule shift mechanisms were suggested to provide more chances for the neighboring nodes to join and to make aggregation schedules in order to maximize the aggregation rate. As a result, the EA protocol has a similar latency and aggregation rate to the DR protocol, but more importantly, it has up to nine times lower energy consumption compared with that of the DR protocol.

References

Donggook Kim received the B.S. degree in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST) in 2000 and the M.S. degree in Electrical Engineering from KAIST in 2002. He is currently a Ph.D. student at KAIST. His research interests include sensor network, multichannel network, localization system, and middleware.

Jaesub Kim received the B.S. degree in Electrical Engineering from Kyungpook National University in 2000, the M.S. degree in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST) in 2002, and the Ph.D. degree in Electrical Engineering from KAIST in 2009. His research interests include sensor network, internet server systems, and operating systems.

Kyu Ho Park received the B.S. degree in Electronics Engineering from Seoul National University, Korea in 1973, the M.S. degree in Electrical Engineering from the Korea Advanced Institute of Science and Technology (KAIST) in 1975, and the Dr.Ing. degree in electrical engineering from the University de Paris XI, France in 1983. He has been a Professor of the Division of EECS, KAIST since 1983. He was a president of Korea Institute of Next Generation Computing in 2005–2006. His research interests include computer architectures, file systems, storage systems, ubiquitous computing, and parallel processing. He is a member of KISS, KITE, Korea Institute of Next Generation Computing, IEEE and ACM.