KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. 11, NO. 5, May. 2017

Reduced-Pipelined Duty Cycle MAC Protocol (RP-MAC) for Wireless Sensor Network

Ngoc Minh Nguyen and Myung Kyun Kim
Department of Computer Engineering, University of Ulsan, Korea
93 Daehak-ro, Nam-gu, Ulsan (44610), Korea
[e-mail: minhnn9889@gmail.com, mkkim@ulsan.ac.kr]
*Corresponding author: Myung Kyun Kim

Received October 26, 2016; revised February 15, 2017; accepted March 9, 2017; published May 31, 2017

Abstract

Recently, the pipeline-forwarding has been proposed as a new technique to resolve the end-to-end latency problem of the duty-cycle MAC protocols in Wireless Sensor Networks (WSNs). Some protocols based on this technique such as PMAC and PRI-MAC have shown an improvement not only in terms of reducing end-to-end latency but also in terms of reducing power consumption. In these protocols, however, the sensor nodes still waste a significant amount of energy for unnecessary idle listening during contention period of upstream nodes to check the channel activity. This paper proposes a new pipeline-forwarding duty-cycle MAC protocol, named RP-MAC (Reduced Pipelined duty-cycle MAC), which tries to reduce the waste of energy. By taking advantage of ACK mechanism and shortening the handshaking procedure, RP-MAC minimizes the time for checking the channel and therefore reduces the energy consumption due to unnecessary idle listening. When comparing RP-MAC with the existing solution PRI-MAC and RMAC, our QualNet-based simulation results show a significant improvement in term of energy consumption.

Keywords: Wireless Sensor networks, pipeline-forwarding, cross-layer, medium access control protocol, routing integrated

---

This work was supported by the 2017 Research Fund of the University of Ulsan.

https://doi.org/10.3837/tiis.2017.05.007 **ISSN : 1976-7277**

1. Introduction

In recent years, with the rapid development of sensor technologies, WSNs are increasingly being applied in many fields, especially in monitoring, such as: air pollution monitoring, forest fire detection, and natural disaster prevention. Even though WSNs and their applications have made strides in the development, the challenges persist and motivate researchers to find better and better solutions. One of the biggest challenges of WSNs is how to prolong the network lifetime for the sensor nodes. Because the sensor nodes are often battery powered and placed in the locations that are hard to reach, it is difficult to change or recharge batteries for them. Reducing energy consumption for sensor nodes becomes a major issue. One of the most important sources that cause the waste of energy is idle listening [1] [2] [3]. It refers to the active listening to an idle channel, waiting for a packet to arrive. During listening time, the sensor nodes do nothing for data transmission but only checking the channel, leading to waste energy.

In order to reduce idle listening, many MAC protocols (e.g., S-MAC [4] [5], T-MAC [6]) have been proposed basing on duty-cycle mechanism. In these protocols, sensor nodes periodically switch their state between active mode and sleep mode. In active mode, sensor nodes perform data communication, while in sleep mode they turn off their radio to save energy. The low duty-cycle based protocols try to let the sensor nodes go to sleep state in as much time as possible. These protocols have shown their efficiency in some simple WSNs.

However, the protocols listed above have some limitations. A major disadvantage of these protocols is that they make the latency in packet delivery increase. For example, in S-MAC, a data packet is only forwarded over one hop for each cycle. The data packets are kept by a sensor node during a sleeping period and have to wait until the next cycle to continue being sent to the sink.

Several approaches such as RMAC [7] and PRMAC [8] had been proposed to address this problem by using cross-layer routing information to reserve a forwarding path, so that enable one or multiple data packets to be transmitted through multiple hops in one cycle. Although these two protocols reduced delivery latency significantly, sensor nodes in the networks that apply these protocols still spend a considerable amount of power for idle listening. This is because all senor nodes have to stay active during the process of reserving forwarding path.

In other approach, some protocols presented the conjunction between routing-integrated and pipeline-forwarding features. With routing-integrated, a MAC protocol can be implemented without another separate routing protocol, leading to reducing control overhead, which is one of the major sources of power consumption. With pipeline-forwarding, not only the idle listening of sensor nodes is shortened, but the delivery latency is also reduced, because the data packets can be transmitted to the sink continuously. Two typical protocols adopt this approach are P-MAC [9] [10] and PRI-MAC [11].

Even though these protocols showed a good result when comparing with some previous solutions, their design still remained some unnecessary idle listening that can be reduced. In these protocols, sensor nodes need to listen to the channel during a quite long duration, in which the upstream node contending with other nodes. In addition, because of using two control frames RTS and CTS separately, a significant amount of power is expended for the handshaking and contending procedures.

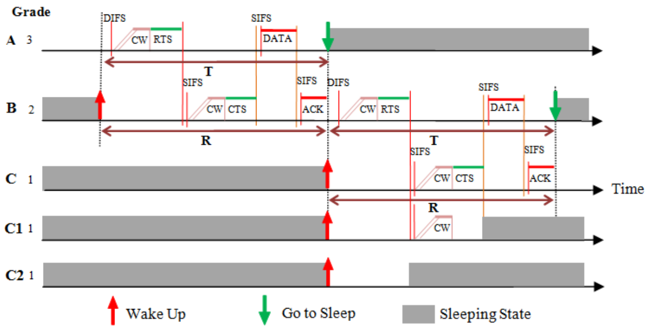


Fig. 1. Overview of a data transmission in PRI-MAC

This paper proposes a new routing-integrated and pipeline-forwarding duty-cycle MAC protocol, named Reduced Pipeline-MAC (RP-MAC), which tries to reduce unnecessary idle listening in recent pipeline-forwarding based protocols. In RP-MAC, sensor nodes are scheduled to wake up right before their upstream nodes send ACK frame. Upon overhearing the ACK, a sensor node can be aware of the possibility of receiving data without waiting for and listening to RTS frame for a long duration. If it cannot overhear any ACK frame from the upstream nodes, it knows that there is no data packet to receive and goes to sleep mode right after that. By this way, the idle listening time can be reduced. Moreover, to further improve energy efficiency, instead of using the traditional RTS/CTS mechanism, RP-MAC uses only one control frame type, named RCTS (Request and/or Clear To Send). This change makes the handshaking procedure shortened and reduces active time of sensor nodes in each cycle. By resorting to Qualnet simulation, we prove that RP-MAC outperforms PRI-MAC in terms of power consumption and end-to-end delay.

The rest of this paper is organized as follows. Section II discusses some related works briefly. Section III presents the detail design of RP-MAC. Section IV gives our performance evaluation result, in comparison with PRI-MAC and RMAC protocols. At last, we conclude the paper.

2. Related Work

The recent MAC protocols based on duty-cycle mechanism can be classified into two categories. The solutions in the first category try to improve performance of network in terms of energy efficiency and/or delivery latency as much as possible, while the second category tries to enhance the quality of service, such as reliability and robustness.

In the first category, the pipeline-forwarding is one of the most effective technique that does not only reduce idle listening time of sensor nodes, but also shorten the time the data packets are kept at the intermediate nodes. D-MAC [12], and PRI-MAC [11] are two representative protocols adopt this technique. In D-MAC, the sensor nodes maintain their wakeup schedules in a staggered manner, that creates a pipeline for data transmission. By this, data packets are transmitted to the sink continuously.

PRI-MAC improved the design of D-MAC by implementing the routing-integrated feature, in conjunction with pipeline-forwarding. This protocol uses the *grade* information of sensor nodes for routing itself. Grade of a node refers to its hop distance from the sink. A data packet is always sent from a higher-grade node to an adjacent-lower-grade node. The sensor nodes in the same grade have to contend for being the next hop. In the starting of network, PRI-MAC performs grade classification, in which the sensor nodes identify their grade. At the same time, the nodes establish their duty-cycle with three states: R (receiving data from upper grade), T (transmitting data to lower grade) and S (sleeping state).

Moreover, PRI-MAC continues to reduce energy consumption for sensor nodes by allowing them to go to sleep state as soon as possible. According to design of PRI-MAC, a node after being rejected from the data transmission in a certain cycle, it goes into the S state early to save energy. In **Fig. 1**, although three nodes C, C1 and C2 have the same grade 1, they go into the S state at the different times. While node C is active during the R state for receiving data from node B after winning the contention, node C1 goes into the S state right after losing. Node C2 even goes into the S state earlier than node C1 because it did not receive the RTS frame from node B.

Nevertheless, the protocols in the first category show their efficiency in only the networks with reliable and stable connection. Otherwise, the collision and retransmission may lead the performance of the network to be very low. The protocols in the second category such as RMS [13] and DSRF [14] address this problem by focusing on providing the reliable methods. For instance, RMS makes the staggered wake up scheduling robust by combining with the multi-parents forwarding scheme.

In this paper, we propose a new solution in the first category, which adopts the routing-integrated and pipeline-forwarding techniques, but further reduces idle listening for sensor nodes.

3. Protocol Design

3.1 RP-MAC overview

The routing-integrated and pipeline-forwarding features have proved their effect in reducing energy consumption and delivery latency. However, in the design of previous protocols that implemented these features, a significant amount of unnecessary idle listening was used for checking the possibility of incoming data packet, leading to waste energy. Especially, in the applications that require light traffic, the waste is being critical, because the data transmission does not often occur.

RP-MAC protocol was designed with the purpose to minimize the idle listening time the sensor nodes need to spend to check whether they should stay active to forward a data packet or not. This is done by overhearing an ACK frame sent by the upper grade node, instead of waiting for a RTS frame for a long period as in the PRI-MAC protocol. In addition to this, RP-MAC shortens the procedures of handshaking and contending for the right to access medium. In RP-MAC protocol, a unique control frame named RCTS is used to replace both RTS and CTS frames used in the traditional MAC protocols.

RP-MAC retains the advantages of cross-layer routing-integrated and pipeline-forwarding features. It means that this protocol uses *grade* of sensor nodes as the key information for directing the transmission of data from higher grade nodes to lower grade nodes. Besides, depending on the grade, each node establishes its duty-cycle schedule, in such a way that the schedules of the nodes in two contiguous grades are overlapped. As mentioned above, however, RP-MAC takes advantage of overhearing to optimize the duty-cycle of the nodes.

A duty-cycle of a sensor node in RP-MAC includes four states: overhearing, receiving, transmitting and sleeping (denoted by O, R, T and S, respectively). In the O state, the sensor nodes overhear the possible ACK frame from the higher-grade nodes. In the R state, the sensor nodes in the same grade contend with each other to receive the data packet from a higher-grade node. In the T state, sensor nodes transmit the data packet to lower grade. And in the S state,

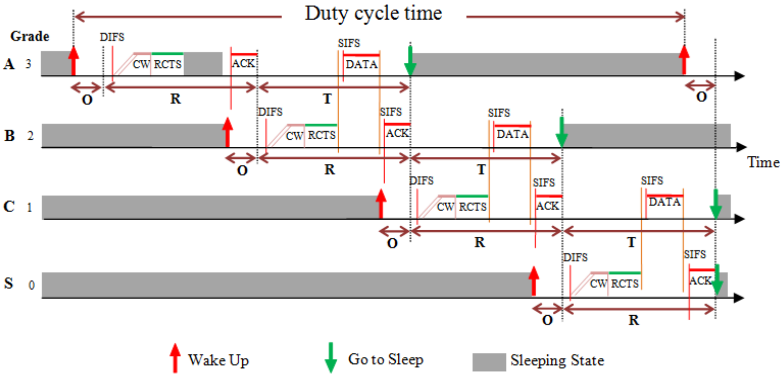


Fig. 2. Data transmission process from source node A to the sink S in RP-MAC

	4 Byte	4 Byte	1Byte	1Byte	4 Byte
Control Frame:	SA	RA	FT	GR	RT

Fig. 3. Format of control frames.

sensor nodes go into sleep mode to save energy. The data transmission in RP-MAC is illustrated in **Fig. 2**.

As introduced above, two sensor nodes in two contiguous grades have staggered schedules. For instance, in **Fig. 2**, node B is in grade 2 and the grade of node C is 1. Right before node B sends ACK frame to node A (at the end of its R state), node C wakes up and overhear that ACK frame (in O state). Node C starts being in R state when node B starts entering T state. The details of the data transmission are given in the sub-section **3.3**.

In RP-MAC design, what makes this protocol improve the performance is the shortening of handshaking procedure to reduce the idle listening time of sensor nodes. For this purpose, RP-MAC implements a new handshaking mechanism using the modified ACK and RCTS control frames, as described in the following sub-section.

3.2 Control frames

Instead of using three conventional control frames: RTS, CTS, and ACK, RP-MAC uses two types of control frames: RCTS and ACK. The common format of these two control frames is illustrated in **Fig. 3**. There are five fields in each frame: SA - address of the node that sends the frame, RA - address of the receiver node, FT - type of control frame - it can be RCTS or ACK, GR - the grade information of the sender, and RT – the relative time from the beginning of current state to the time the sender generates this control frame.

RCTS is the combination of the traditional RTS and CTS. It means that the RCTS frame has both two responsibilities: the first is notifying other sensor nodes that the sender has a data packet to send; and the second is to notify of being ready to receive data packet. For example, in **Fig. 2**, node A is the source node that has a data packet to send to the sink. In R state, after winning the contention and getting the right to use the channel, node A sends an RCTS to declare that it is the winner in the contention. In this case, the RCTS keeps function of RTS and the RA field in the frame is set to the broadcast address. In the case of node B, the RCTS frame is used to notify node A that it is ready to receive data packet. The other nodes in the same grade will go to sleep state right after overhearing this RCTS frame from B. In this case, the RCTS frame keeps function similar to the CTS, and the RA field is set to the address of node A.

ACK frame is also used for two purposes. The first purpose is to acknowledge the receipt of data packet from the higher-grade node; and the second is to notify the lower-grade nodes of the availability of data. If a sensor node overhears an ACK frame from a higher-grade node, it knows that this node is going to send a data packet. In contrast, if the sensor node does not overhear any ACK frame during O state, it goes into the S state immediately.

3.3 Data transmission

This sub-section describes a data transmission from a source node to the sink, which is illustrated in **Fig. 2**. In this figure, the source node A wants to send its data packet to the sink S. During O state, it does not overhear ACK frames from higher-grade nodes. In R state, this node contends with other nodes in the same grade to get the right to use the channel. Follows the IEEE 802.1, node A chooses a random back-off value for the contention window and starts counting down. It is supposed that the random back-off value of node A is the smallest, after counting down to 0, it broadcasts an RCTS frame. Other nodes in the same grade will go into the S state after overhearing this RCTS. Since node A does not receive any data from the higher-grade nodes, it also turns off radio and sleeps for a short time, but wakes up at the end of R state to send an ACK frame. The purpose of this sending is only to notify the lower-grade nodes that it is going to send a data packet. The RA field in this ACK frame is also set to the broadcast address.

As mentioned previously, after the initialization process, each sensor node has been scheduled in such a way that it can overhear ACK from its higher-grade nodes right after it wakeups. In the example in **Fig. 2**, node B wakes up after the S state and goes into the O state right before the time node A sends its ACK frame. By overhearing this ACK, node B can know that node A is going to send a data packet. When node A starts entering its T state, node B also starts R state. Since some neighboring nodes in the same grade of node B may also overhear the ACK frame of node A, they have to contend with each other to become the next forwarder of the data packet by starting a contention window at the beginning of the R state. Assume that node B wins the contention, it sends an RCTS frame to node A immediately. The other contenders, which are still counting down their back-off value can overhear this RCTS frame and go to the S state right after that. Upon receiving the RCTS from B, node A starts sending its data packet. After the transmission from A to B finishes, node B does not reply node A immediately but sends an ACK frame at the end of the R state to make sure that the lower-grade nodes can overhear this frame. This process is repeated until the data packet reaches the sink.

3.4 State duration

In this part, we give the formulas to calculate the duration of each state in one cycle of RP-MAC, as well as how to choose a good value for the durations when deploying the protocol.

R/T state: As shown in **Fig. 2**, the duration of the T state of a sensor node is equal to the duration of R state of its down-stream node, thus these two states have the same duration, which is calculated by the following formula:

$$T_{R/T} = DIFS + 2SIFS + CW_{MAX} + durRCTS + durDATA + durACK \qquad (1)$$

Where, *durRCTS*, *durDATA* and *durACK* are the transmission duration of RCTS, DATA and ACK, respectively; CW_{MAX} is the maximum duration of contention window.

O state: In RP-MAC, duration of the overhearing state needs to be long enough for a node to receive an ACK frame. It can be determined by the following formula:

$$T_O = SIFS + durACK \tag{2}$$

In real WSNs, since the clock rates of the sensor nodes may not be the same and clocks of sensor nodes can even drift apart over time, to ensure that the period of O state covers the ACK transmission, the sensor nodes should be scheduled to wake up a small time earlier than the calculated time.

S state: The duration of sleeping state is the rest of one cycle duration (T_{cycle}) and is computed as follow:

$$T_S = T_{cycle} - (2T_{R/T} + T_O) \tag{3}$$

The duration of one cycle needs to be chosen to guarantee that the sensor nodes sleeps long enough to avoid the interference among grades. For example, in **Fig. 2**, assume that the interference range is about twice as long as transmission range, if node C (grade 1) is in the T state while node A (grade 3) is in the R state, the collision may occur. Node C should go into the S state before node A goes to the R state. In other words, the following inequality needs to be satisfied:

$$T_S + T_O \geq 2T_{R/T} \tag{4}$$

or,

$$T_{cycle} \geq 4T_{R/T} \tag{5}$$

3.5 Grade identification and schedule establishment

Two main factors that make the network operate properly following the design of RP-MAC are grade information of sensor nodes and their relevant and accurate schedules. As introduced in sub-section **3.1**, the process of identifying grade and establishing schedule for sensor nodes take place during the initialization process at the start of network. Establishing schedule refers to the action that a sensor node determines which state (O, R, T or S) it should be being in, at a certain time, and how long after that it switches to the next state. This part describes detail about this process.

After being distributed in the network area and then started, all sensor nodes stay in active mode during a predefine duration for initialization. This process includes two periods. First of all, the clocks of sensor nodes need to be synchronized to the same origin of time. In the first period, it is done by using a separate protocol. Sub-section **3.7** will present more detail about this problem.

In the second periods, a message named INIT is broadcasted to all the sensor nodes in the network by the flooding technique. Based on the information in INIT message, the sensor nodes can identify their grade and after that, establish the corresponding schedule of operation. The next part describes detail of the steps that the sensor nodes work.

- Grade initialization: the sink sets its grade to 0; all other nodes in the network set their grade to -1 as a default value.
- The sink establishes its schedule by checking the current time, determining which state it should be being in and how long it should change to the next state. It is conventional that at the origin of time, the sink starts entering O state.
- The sink generates and then broadcasts an INIT message which contains the necessary information for other nodes to identify the grade and establish their schedule,

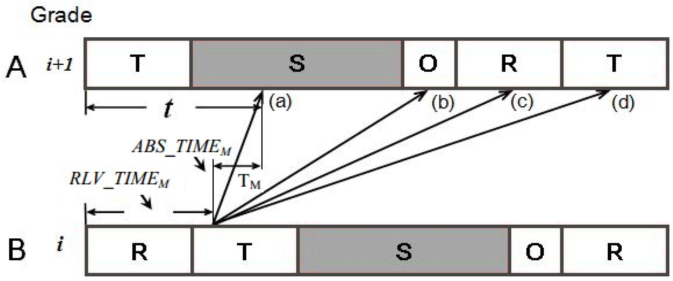


Fig. 4. An example of grade identification and schedule establishment.

including:

  - $GRADE_M$, the grade of the sending node.
  - RLV_TIME_M, the relative time of the sending node, which is measured from the beginning of the R state of current operational cycle to the generation time of the message as shown in **Fig. 4**.
  - ABS_TIME_M, the absolute time of the sending node at the time the INIT message is generated.
- A node on receiving an INIT message may set its grade and establish its schedule based on the information in the received message.
- The node then continues to generate and broadcast an INIT message of itself.

This process is iterated until the end of initialization period.

Algorithm 1 gives the pseudo-code of how a sensor node handles an INIT message received from other nodes. In the pseudo-code, $GRADE_N$ denotes the grade information of sensor node at the time before receiving the INIT message.

Need to note that a sensor node can receive INIT message more than one time. As shown in the **Algorithm 1**, a node will update its grade information and establish schedule in only two cases. The first case is that it has not joined any grade yet (current grade $GRADE_N = 1$), the node then sets up initialization information for the first time. Another case is that the node has already joined a grade, but the new grade will be better than the current one (a grade is called better than another for a node if it is smaller, or in other words, nearer from the sink than the other). In other cases, the node discards the received INIT message. After setting grade information, the sensor nodes establish or update their schedule according to the **Algorithm 2**.

When implementing RP-MAC, four states R, T, S and O are numbered from 0 to 3 in order. Each sensor node maintains a *DURATION* array which stores the duration of four states (calculated in sub-section **3.4** above). For example, the array member *DURATION [0]* stores the duration of the R state and the array member *DURATION [3]* stores the duration of the O state. **Algorithm 2** tries to find out the state that a sensor node should be in at the time it receives an INIT message.

The **Fig. 4** provides an example for analysis. After receiving an INIT message from node B, node A calculates the delay time (T_M) from the time when node B generated the INIT message to the moment node A receives it. Because at the time node B goes to R state, node A should go to T state, ($RLV\_TIME_M + T_M$) - denoted by t, is the relative time of node A measured from the beginning of one previous T state to the time of receiving the message. By subtracting state durations one by one from t until getting a negative value, node A can be aware of which state it should be in and how much time it passed in that state at current moment. For example, in case (a) of **Fig. 4**, t is firstly decreased by duration of the T state. After being decreased, t is still greater than 0. It means the node A has already passed through the T state. t is

Algorithm 1 Handling INIT message

```
if $GRADE_N < 0 \,||\, GRADE_N > (GRADE_M + 1)$ then
    $GRADE_N \leftarrow GRADE_M + 1$
    establish/update the schedule using Algorithm 2
    $GRADE_M \leftarrow GRADE_N$
    $RLV\_TIME_M \leftarrow$ calculate relative time
    $ABS\_TIME_M \leftarrow$ get time of system
    choose the random backoff time
    rebroadcast the INIT message
else
    discard the INIT message
end if
```

Algorithm 2 Establish/Update Schedule

```
$T_M \leftarrow$ system time $- ABS\_TIME_M$
$t \leftarrow RLV\_TIME_M + T_M$
$state \leftarrow 0$
while $t > 0$ do
    $state \leftarrow (state + 1) \% 4$
    $t \leftarrow t - DURATION[state]$
end while
$STATE_N \leftarrow state$
set timer to go to the next state $\leftarrow |t|$
```

continuously decreased by duration of the S state. This time, t becomes less than 0. Node A infers that it is currently in the S state and after a duration of $|t|$, node A will go to the next state (O state).

Similarly, we can analyze the cases (b), (c) and (d). As written in the **Algorithm 1**, after establishing its schedule, node A updates three fields: $GRADE_M$, RLV_TIME_M and ABS_TIME_M in its own INIT message. To avoid collision with INIT messages from other nodes, node A does not rebroadcast its message immediately but waits for a random back-off time.

During the initialization process, all sensor nodes in the network are active, even in sleeping state.

3.6 Handling frame loss

Because the sensor nodes in RP-MAC are not scheduled to receive the retransmitted packet, no retransmissions are performed in the same cycle. If a sensor node does not receive the ACK frame from its downstream node, it will keep the data frame, goes into sleep state and resend it in the next cycle.

With RCTS frame, the sensor nodes use a timer to count up the time it has been waiting for the RCTS frame. This timer is started at the beginning of the T state with the value of interval is $(DIFS + CW_{MAX} + durRCTS + SIFS)$. It will be canceled when the sensor node receives the expected RCTS. Otherwise, when the timer expires, the node immediately goes to sleep state to save energy.

For example, in **Fig. 2**, if node A fails to receive the RCTS or the ACK frame sent from node B, it keeps the data packet, goes to the S state and will try to contend for the right to forward it in the next cycle. The packet will arrive to the sink late at least the duration of a cycle.

However, the impact of a frame loss on the average end-to-end delay is depended on each case. Because in each cycle, the sink can receive only one data packet, some time a frame loss makes other data packet forwarded to the sink earlier. In the best case, a frame loss does not affect the average end-to-end delay. For example, if two packets P1 and P2 are on their way forwarding to the sink at the same time. Until a certain time, the sensor nodes that are holding them have to contend with each other. In normal case, if the node holding P1 wins the contention, P2 needs to wait until the next cycle to be sent. In case of frame loss, assume that forwarding P1 is delayed, so P2 can be forwarded to the sink in current cycle. Finally, the average latency is not changed. However, in the worst case, a frame loss may lead to delay of all other data packets.

3.7 Synchronization

Synchronization is a big problem in WSNs. In a network, each node has its own local clock for determining the timing of the events. In order to be able to collaborate, all sensor nodes need to maintain the same time scale. The synchronization aims to guarantee this. In the WSNs that implement RP-MAC protocol, synchronization problem is resolved by two different methods for the initialization process and during the operation period after that.

In the initialization process of network, the synchronization is required before performing schedule establishment, so that all the nodes have the same reference point of origin time. This can be done by exploiting a separate synchronization protocol (e.g., [15], [16], [17], and [18]). The sensor nodes run with such a protocol during a duration long enough to make sure that all of them have a required precise clock.

During the operation period of network, in order to correct the schedule error caused by the factors such as clock drift and variable hardware/OS latency, RP-MAC uses a loose synchronization scheme introduced in P-MAC and PRI-MAC. In control frames, a four-byte field named RT (relative time) is added. It indicates the duration from the beginning of current state of the sender node to the time that the node generates the frame. In other words, it is how long the node has been in that state. By overhearing the control frames sent from the neighbor nodes, a sensor node can use this information to adjust its clock.

For instance, if node A in grade i^{th} receives an RCTS frame from node B in grade $(i\text{-}1)^{th}$, based on the value of RT field in the frame, node A calculates how long it should have been in current T state, as the following formula:

$$t = RT + durRCTS \tag{6}$$

Depending on the value of t, node A can shorten or extend the duration of incoming S state, so that it can start the next cycle at correct time.

If a node has large time error and loses synchronization, it may not receive any frame during a certain time or receives a frame at a wrong state. In these cases, it keeps active for a period and listens to channel to revise and adjust its grade and/or schedule.

4. Performance Evaluation

4.1 Theoretical evaluation

In this part, some theoretical comparisons between RP-MAC and PRI-MAC will be given to evaluate the performance of RP-MAC. The comparisons are performed base on three metrics: idle listening, per-hop delay, and control overhead.

In these comparisons, a same value of each parameter is used for both protocols. These parameters include the cycle duration, size of frames (RCTS, ACK, RTS and CTS), bandwidth, maximum back-off value and time slot duration. All control frames have the same size, so the times needed for transmitting them are also the same. In the next formulas, *durCTRL* will be used as a common duration time of all the followings: *durRCTS*, *durACK*, *durCTS* and *durRTS*. The duration times of the states of PRI-MAC are calculated according to [11] based on the assumption of no collision or retransmission. This assumption is applied fairly to all comparing protocols.

1) Idle listening: This part gives the comparison of idle listening time that a sensor node spends in each operational cycle, between two protocols. Idle listening time of a sensor node is considered as the time it spends in active mode but not for transmitting, receiving or overhearing any data or control frame. In the sending of one data packet from a sensor node to the sink, a certain sensor node in the network can be one of the following types:

- (i) Source node: the node that starts the data sending.
- (ii) Forwarding nodes: the intermediate nodes that forward the data packet (each grade has only one forwarding node for one data packet).
- (iii) Contending nodes: the nodes that participate but lose the contention to forward the data packet (each grade may have no, one or more than one contending nodes).
- (iv) Disjoining nodes: the nodes which cannot listen to the control frame (RTS frame in PRI-MAC and ACK frame in RP-MAC) from the forwarding node in the higher-grade (each grade may have no, one or more than one disjoining nodes).
- (v) Receiver node: the destination of the data transmission. The receiver node goes to sleep state right after sending the ACK frame to its upstream node.

Need to note that, even though the type of sensor nodes may change after each cycle depending on the data transmission, in the networks with light traffic, a sensor node is disjointing type in most of cycles. Moreover, the number of disjointing nodes in each cycle is normally much greater than the number of nodes in other types.

The following parts analyze the idle listening time of each type of nodes.

a) Source node and Forwarding node: The forwarding nodes have the longest active time in comparing to other nodes. In one operational cycle of RP-MAC, a forwarding node overhears one ACK frame, sends one RCTS frame, one ACK frame and one DATA packet; it also receives one RCTS frame, one ACK frame and one DATA packet. Therefore, the idle listening time of a forwarding node in RP-MAC is given by the follows:

$$\begin{aligned} T_{idl\_fn}(RP-MAC) &= T_O + 2T_{R/T}(RP-MAC) - (5durCTRL + 2durDATA) \\ &= 2DIFS + 5SIFS + 2CW_{MAX} \end{aligned} \tag{7}$$

In RP-MAC protocol, since a source node does not need to receive a data packet from its higher-grade node, it sleeps instead. Therefore, the idle listening time of a source node is equal to that of a forwarding node.

$$T_{idl\_srn}(RP-MAC) = 2DIFS + 5SIFS + 2CW_{MAX} \tag{8}$$

The idle listening time of a forwarding node and a source node in PRI-MAC are also the same and given by the follow:

$$T_{idl\_srn}(PRI-MAC) = T_{idl\_fn}(PRI-MAC) = 2DIFS + 6SIFS + 4CW_{MAX} \tag{9}$$

The difference between idle listening time of a forwarding node (and of a source node) in RP-MAC and PRI-MAC can be calculated as follows:

$$T_{idl\_fn}(PRI-MAC) - T_{idl\_fn}(RP-MAC) = SIFS + 2CW_{MAX} \tag{10}$$

b) Contending nodes: In RP-MAC protocol, the contending nodes go to sleep state right after overhearing the RCTS frame from the forwarding node of the same grade. Their idle listening time in one cycle is given by the follow:

$$T_{idl\_cn}(RP-MAC) = SIFS + DIFS + CW_{fn} \tag{11}$$

where, CW_{fn} is the duration of contention window of the forwarding node in the same grade.

In PRI-MAC protocol, the contending nodes also go to sleep state right after overhearing the CTS frame from the forwarding node of the same grade.

$$T_{idl\_cn}(PRI-MAC) = DIFS + CW_{MAX} + SIFS + CW_{fn} \tag{12}$$

The difference between idle listening time of a contender node in RP-MAC and PRI-MAC can be calculated as follows:

$$T_{idl\_cn}(PRI-MAC) - T_{idl\_cn}(RP-MAC) = CW_{MAX} \tag{13}$$

c) Disjoining nodes: In RP-MAC, these nodes go to sleep state right after O state. They spend all time of O state for idle listening, so:

$$T_{idl\_dn}(RP-MAC) = T_O = DIFS + durCTRL \tag{14}$$

In PRI-MAC, these nodes go to sleep after could not overhear the RTS frame from the forwarding node of the higher-grade. The idle listening time of them is given by the follow:

$$T_{idl\_dn}(PRI-MAC) = DIFS + CW_{MAX} + durCTRL \tag{15}$$

The difference between idle listening time of these nodes in RP-MAC and PRI-MAC can be calculated as follows:

$$T_{idl\_dn}(PRI-MAC) - T_{idl\_dn}(RP-MAC) = CW_{MAX} \tag{16}$$

d) Receiver node: The idle listening time of this node in two protocols are calculated as follows:

$$T_{idl\_rn}(RP-MAC) = DIFS + 3SIFS + CW_{MAX} \tag{17}$$

$$T_{idl\_rn}(PRI-MAC) = DIFS + 3SIFS + 2CW_{MAX} \tag{18}$$

The difference between idle listening time of the receiver nodes in RP-MAC and PRI-MAC is given by the follow:

$$T_{idl\_rn}(PRI-MAC) - T_{idl\_rn}(RP-MAC) = CW_{MAX} \tag{19}$$

Since the time of contention windows is a long duration in comparing to the time of sending control frames, it is clear to see that all types of nodes in RP-MAC have significant shorter idle listening times than that of PRI-MAC.

2) Control overhead: Protocol overhead is one of the reasons that lead to energy waste in WSNs [19] [20]. It refers to the frame headers and the signaling required by the MAC protocol [21]. Due to the using of the same packet header and control frame size, the different of control overhead between RP-MAC and PRI-MAC can be evaluated by comparing the number of control frames used for sending a data packet from a source node to the sink.

Table 1. Comparison of number of control frame

Type of node	RP-MAC	PRI-MAC
Source	2 (RCTS, ACK)	1 (RTS)
Forwarding nodes	2 (RCTS, ACK)	3 (RTS, CTS, ACK)
Sink	2 (RCTS, ACK)	2 (CTS, ACK)
Total	**4+2*n**	**3+3*n**

Table 1 shows the number of control frames that is generated by three types of sensor node on a data-forwarding path (source node, forwarding nodes and the sink) in two protocols. In the table, **n** denotes the number of forwarding nodes on the path. It is clear to see that when **n** is greater than one, RP-MAC uses less control frames than PRI-MAC. Moreover, the more number of relaying nodes is (or the wider the network is), the better RP-MAC reduces control overhead.

3) Per-hop delay: The per-hop delay refers to the time a data packet stays in each hop. This part considers per-hop delay in only the case without retransmission. In RP-MAC and PRI-MAC, they are calculated as follows:

$$T_{phd}(RP-MAC) = 2durCTRL + CW_{MAX} + DIFS + 2SIFS \qquad (20)$$

$$T_{phd}(PRI-MAC) = 3durCTRL + 2CW_{MAX} + DIFS + 3SIFS \qquad (21)$$

The difference of basic per-hop delay between RP-MAC and PRI-MAC can be calculated as follows:

$$T_{phd}(PRI-MAC) - T_{phd}(RP-MAC) = durCTRL + SIFS + CW_{MAX} \qquad (22)$$

It is also clear to see that RP-MAC reduces per-hop delay.

4.2 Simulation evaluation

1) Experiment setup: The performance of RP-MAC protocol is evaluated experimentally by using the network simulation QualNet 5.02 [22] in comparison with two existing duty-cycle based MAC solutions, RMAC and PRI-MAC.

Table 2, **Table 3** and **Table 4** show the key networking, frame size and cycle duration parameters used. The transmission power of each node is set to -24dBm, which is approximately 25 meters of transmission range. The maximum length of the contention window is set with 64 slots. Duration of each slot is 0.320ms, which is chosen according to IEEE 802.15.4 standard, together with the values of DIFS and SIFS.

Table 2. Networking parameters

Parameters	Value	Parameters	Value
Transmission power	-24 dBm	Contention Window (CW)	20.48 ms
Sensor energy model	MicaZ	DIFS	0.832 ms
Path loss model	2-ray	SIFS	0.192 ms

Table 3. Frame type parameters

Frame types	Frame size (bytes)	Frame types	Frame size (bytes)
RTS/CTS	10	DATA	128
RCTS	10	INIT	10
ACK	10	PION	14

Table 4. Cycle duration parameters

	$T_{cycle}\,(ms)$	$T_{R/T}\,(ms)$	$T_{O}\,(ms)$	$T_{S}\,(ms)$
RP-MAC	1000	27.736	0.968	943.56
PRI-MAC	1000	49.184	-	901.631
	T_{cycle} (ms)	T_{DATA} (ms)	T_{SYNC} (ms)	T_{SLEEP} (ms)
PR-MAC	1000	38	12	950

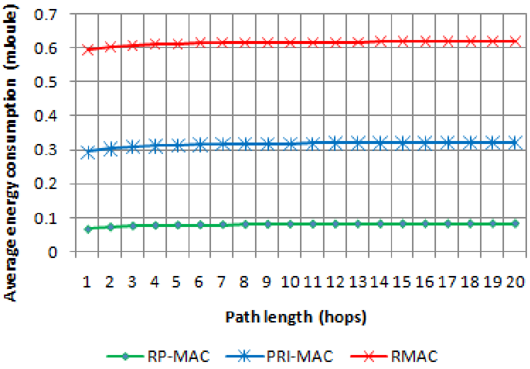


Fig. 5.. Average energy consumption per node vs. path length

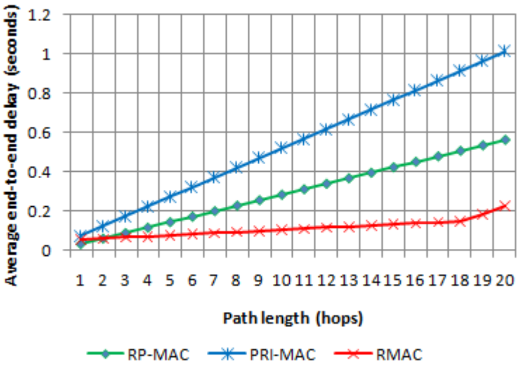
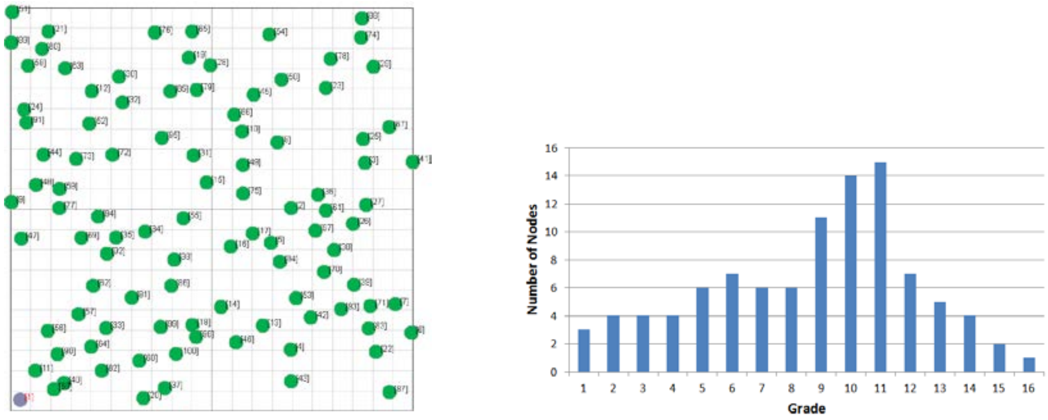


Fig. 6. Average end-to-end latency vs. path length

Simulations were conducted with two types of network prototype: the chain network and the random network. In all simulations, we assume that the clock times of all nodes in the network are synchronized according to the clock time on simulation tool. The following parts provide the results of the evaluation in each network prototype. In each part, two metrics were used to compare the performance of protocols: energy consumption and end-to- end delay.

2) Simulation result with chain network topology: In our simulations with chain network, all nodes are evenly positioned on a straight line. The distance between two neighboring nodes is 20 meters. The source node sends data to the sink using CBR (constant bit rate) flow at a rate of 1 packet every 10 seconds. The length of chains varies from 1 to 20 hops. Using the chain topology is for studying the basic multi-hop delivery of the protocols.

Fig. 5 and **Fig. 6** show the difference among three protocols, in terms of the average energy consumed by sensor nodes and the average end-to-end delay time of data packets. As can be seen from two figures, even though RMAC is the most effective protocol of reducing delivery latency, the energy expended by the sensor nodes in this protocol is much greater than that in RP-MAC and PRI-MAC. This is because in RMAC, forwarding path is reserved before the transmission takes place. The data packets are transmitted through the sensor nodes continuously, without waiting for contending and handshaking procedures. However, all nodes have to be active during period of reserving path, leading to waste a very big amount of energy. Whereas, sensor nodes in RP-MAC consume the least power, while this protocol still guarantees a good performance in terms of delivery latency. **Fig. 6** indicates that the average energy consumption in RP-MAC is more three times lower than that of PRI-MAC and the ratio is more six times lower when comparing to RMAC. This is the result of minimizing idle listening for sensor nodes of RP-MAC.



a. Node distribution

b. The histogram of the path lengths

Fig. 7. Random network topology

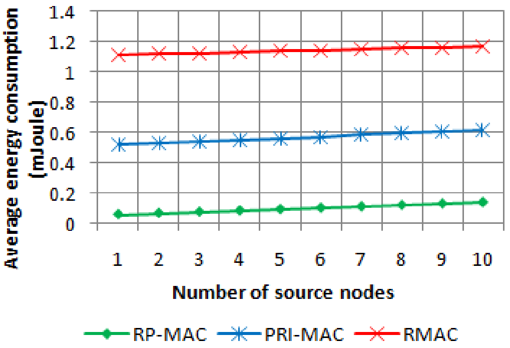


Fig. 8. Average energy consumption in random network

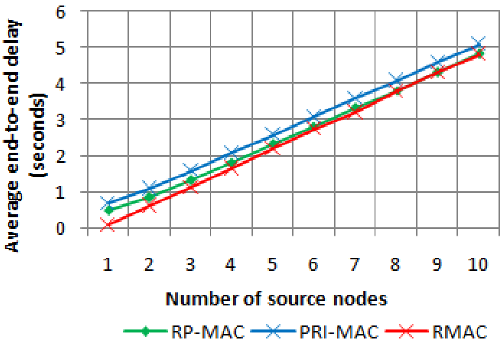
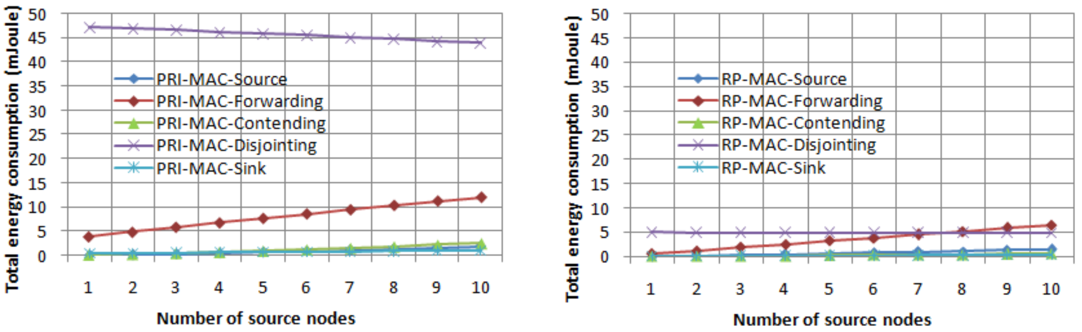


Fig. 9. Average end-to-end latency in random network

3) Simulation results with random network topology: **Fig. 7a** shows an example of a network with 100 sensor nodes distributed randomly in a square area of 200m x 200m. The sink node is located at the bottom-left corner. Refer to **Fig. 7b**, which shows the node distribution according to the hop distance to the sink, we can see that maximum value of the grade of sensor nodes is 16. In the simulations with this type of network, the traffic is generated as follows. A predefined number of sensor nodes are randomly selected as sources to generate data packets at the same time. Each source generates one packet.

To evaluate performance of RP-MAC in overview, we carried out 10 simulations with the network displayed in **Fig. 7a**, and then measured the average energy consumption and average end-to-end delay in three protocols. In these simulations, the number of random source nodes varies from 1 to 10. The source nodes generate data periodically with interval of 20 seconds. In each simulation, network operates for two hours.

Fig. 8 presents the comparison of the average energy consumption of sensor nodes in RP-MAC with PRI-MAC and RMAC in random network topology. The result in this topology is similar to the result in the chain topology, which had proved that RP-MAC is the protocol with the best energy efficiency. A different point between two results is that the energy consumption in all three protocols increases steadily according to the number of packets



a. Energy consumption in PRI-MAC protocol

b. Energy consumption in RP-MAC protocol

Fig. 10. Total energy consumed by different node types in random network

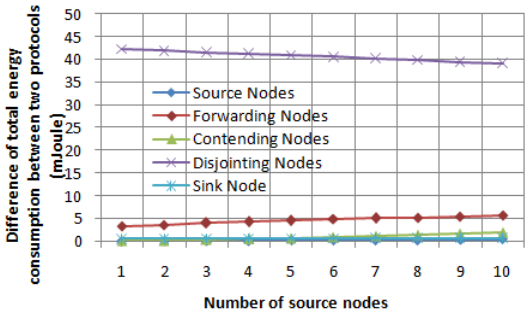


Fig. 11. Energy consumption reduced by RP-MAC in different node types

generated at the same time. This is because in RP-MAC and PRI-MAC, the more the number of packets transmitted at the same time is, the more number of forwarding nodes and contending nodes are, whereas the less number of disjoining nodes is. Since the forwarding nodes and contending nodes consume more energy than the disjoining nodes do, it leads to the average of energy consumption increases in both protocols. The reason for this in RMAC is similar.

Fig. 9 shows the average packet end-to-end delays by RP-MAC, PRI-MAC and RMAC, according to the variation of traffic density. The latencies of all three protocols increase with the increase of number of data packets generated at the same time. It makes sense because the more generated packets are, the more contention is. Especially, the contention is high at sensor nodes located one or two hops away from the sink.

The result displayed in **Fig. 9** indicates that in random network, the improvement of RP-MAC in term of end-to-end delay is just slight when comparing with the result in the chain network. This can be explained as the flowing reason. In our simulation, the data packets are generated periodically, but the source nodes are chosen randomly. Since the sensor nodes in different grades maintain different schedules, two data packets that are generated at the same time may start being sent to the sink at the different time. In RP-MAC, if a data packet is generated at the moment that the source node is in the R, T or S states, it has to wait until next cycle to start being sent. In PRI-MAC, however, if a data packet is generated at the beginning of R state (before sub-state of contention for CTS), it still can be sent in the same cycle. This reduces the difference of end-to-end delay between to protocol.

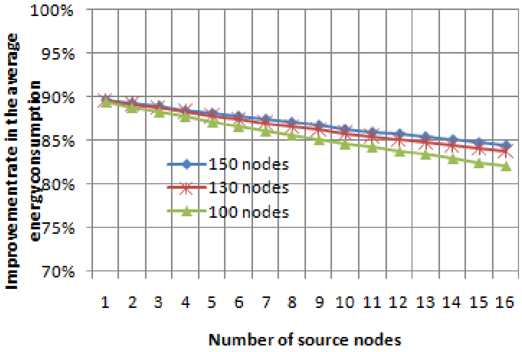


Fig. 12. Ratio of energy consumption reduced by RP-MAC from PRI-MAC in different networks

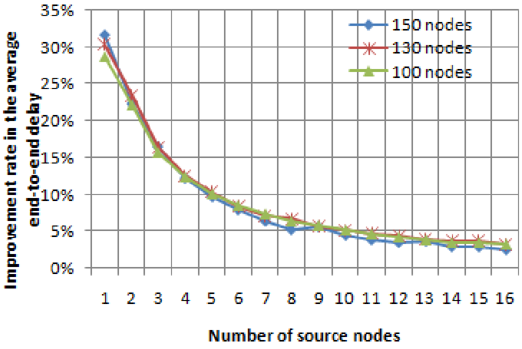


Fig. 13. Ratio of end-to-end delay reduced by RP-MAC from PRI-MAC in different networks

In order to evaluate more deeply the energy consumption in two pipeline-forwarding protocols RP-MAC and PRI-MAC, we analyze how much energy each type of nodes in the network uses. **Fig. 10** shows the energy consumed in PRI-MAC and RP-MAC by different kinds of the sensor node after the simulation runs two hours. The energy consumption of each node type is accumulated after every cycle. Because there is only one sink in the network and the sink is the unique receiver node of all the data transmissions, the power consumed by it is calculated separately, even this node can be disjoining type or receiver type.

As can be seen in **Fig. 10**, two types of node that consume the most power are the disjoining and forwarding types. This is because the disjoining type has the largest number of nodes in the network, while the forwarding type uses much more energy than other types. However, in PRI-MAC, power consumed by disjoining nodes dominates others, whereas, in RP-MAC, the differences in power consumption between disjoining type and other types are not so great. In RP-MAC, when the packet rate is higher than seven packets per twenty seconds, the disjoining nodes consume energy even less than that by the forwarding nodes.

Fig. 11 shows the amount of energy that RP-MAC reduces for each type of nodes, from PRI-MAC. It is easy to see that the improvement almost comes from reducing energy consumption of the disjoining nodes.

Finally, we investigated the impact of number of sensor nodes on performance of network. We repeated the experiments above in two other networks, in which 130 and 150 sensor nodes are distributed, respectively. From the results of average energy consumption and average end-to-end delay of two protocols RP-MAC and PRI-MAC, we measured the improvement ratio that RP-MAC did in comparing with PRI-MAC. In other words, it is how many percent of energy consumption and delivery latency RP-MAC reduces from PRI-MAC. The results are displayed on **Fig. 12** and **Fig. 13**.

Fig. 12 shows the improvement of RP-MAC in terms of energy consumption. As can be seen from the figure, the more number of sensor nodes in network, the more amount of power consumption is reduced. This can be explained by the ratio of disjointing node. It is easy to see that in the networks with light traffic, this ratio increases together with the increase of total number of sensor nodes, leading to reduce average energy consumption. The figure also indicates that the improvement decreases when the traffic becomes heavier. However, when the number of source nodes reaches value of 16, the improvement ratio is still high with more than 80%. It is a significant improvement.

Fig. 13 shows the improvement of RP-MAC in terms of end-to-end delay. It is clear to see that the rate of delay reducing rapidly decreases when the traffic is higher. However, it is almost the same in all three networks. In other words, the improvement rate of end-to-end delay in RP-MAC is not depended on the number of sensor nodes in network.

5. Conclusion

This paper presented the design of RP-MAC - a new pipeline-forwarding duty-cycle MAC protocol for WSNs. RP-MAC mitigates the handshake procedure by taking advantage of ACK mechanism and using a new control frame RCTS, instead of the traditional RTS and CTS frames. By this way, RP-MAC reduces the idle listening time of sensor nodes. Moreover, with the routing-integrated feature, RP-MAC enables the data packets to be forwarded continuously from the source node to the sink without using any specific routing protocol. Our simulation results shown that RP-MAC archives better performance than two previous MAC protocols PRI-MAC and RMAC in terms of reducing power consumption and end-to-end delay. Our protocol can be potentially used for energy-sensitive applications with light traffic.

6. Disclosure

Part of this work was published in the International Wireless Communications and Mobile Computing Conference, IWCMC 2015, Dubrovnik, Croatia, August 24-28, 2015 [23].

References

[1] L. M. Feeney and M. Nilsson, “Investigating the energy consumption of a wireless network interface in an ad hoc networking environment,” in *Proc. of IEEE INFOCOM*, pp. 1548-1557, 2001. Article (CrossRef Link).

[2] E. Shih, P. Bahl, and M. J. Sinclair, “Wake on wireless: an event driven energy saving strategy for battery operated devices,” in *Proc. of ACM MobiCom*, pp. 160-171, 2002. Article (CrossRef Link).

[3] I. Demirkol, C. Ersoy, and F. Alag oz, “MAC protocols for wireless sensor networks: a survey,” in *Proc. of IEEE Communications Magazine*, vol. 44, no. 4, pp. 115-121, 2006. Article (CrossRef Link).

[4] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” in *Proc. of IEEE INFOCOM 2002*, pp. 1567-1576, 2002. Article (CrossRef Link).

[5] W. Ye, J. Heidemann, and D. Estrin, “Medium access control with coordinated adaptive sleeping for wireless sensor networks,” in *Proc. of IEEE/ACM Transactions on Networking*, vol. 12(3), pp. 493-506, Jun. 2004. Article (CrossRef Link).

[6] T. van Dam and K. Langendoen, “An adaptive energy-efficient MAC protocol for wireless sensor networks,” in *Proc. of SenSys 2003*, pp. 171-180, Nov. 2003. Article (CrossRef Link).

[7] S. Du, A.K. Saha, and D. B. Johnson, “R-MAC: a routing-enhanced duty-cycle MAC protocol for wireless sensor networks,” in *Proc. of IEEE INFOCOM*, pp. 1478-1486, May 2007. Article (CrossRef Link).

[8] T. Canli and A. Khokhar, “PRMAC: Pipelined Routing Enhanced MAC Protocol for Wireless Sensor Networks,” in *Proc. Communications, 2009. ICC ’09. IEEE International Conference*, pp. 1-5, Jun. 2009. Article (CrossRef Link).

[9] F. Tong, R. Xie, L. Shu, and Y. Kim, “A cross-layer duty cycle MAC protocol supporting a pipeline feature for wireless sensor networks,” *Sensors*, 11(5): 5183-5201, 2011. Article (CrossRef Link).

[10] F. Tong, W. Tang, R. Xie,L. Shu,and Y. Kim, “P-MAC: A cross-layer duty cycle MAC protocol towards pipelining for wireless sensor networks,” in *Proc. of IEEE ICC*, pp. 1-5, 2001. Article (CrossRef Link).

[11] Fei Tong, Minming Ni, Lei Shu and Jianping Pan, “A Pipelined-forwarding, Routing-integrated and effectively-Identifying MAC for large-scale WSN,” in *Proc. of IEEE GLOBECOM*, pp. 225-230, 2013. Article (CrossRef Link).

[12] G. Lu, B. Krishnamachari, and C.S. Raghavendra, “An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks,” in *Proc. of International Parallel and Distributed Processing Symposium*, pp. 224-231, Apr. 2004. Article (CrossRef Link).

[13] Yongle Cao, Shuo Guo and Tian He, “Robust multi-pipeline scheduling in low-duty-cycle Wireless Sensor Networks,” in *Proc. of IEE INFOCOM*, pp. 361-369, 2012. Article (CrossRef Link).

[14] Long Cheng, Yu Gu, Tian He and Jianwei Niu, “Dynamic switching-based reliable flooding in low-duty-cycle wireless sensor networks,” in *Proc. of IEEE INFOCOM*, pp. 1393-1401, 2013. Article (CrossRef Link).

[15] J. Elson, L. Girod, and D. Estrin, “Fine-grained network time synchronization using reference broadcasts,” in *Proc. of Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002)*, pp. 147-163, Dec. 2002. Article (CrossRef Link).

[16] S. Ganeriwal, R. Kumar, and M. B. Srivastava, “Timing-sync protocol for sensor networks,” in *Proc. of First International Conference on Embedded Networked Sensor Systems (SenSys 2003)*, pp. 138-149, Nov. 2003. Article (CrossRef Link).

[17] M. Marti, B. Kusy, G. Simon, and. Ldeczi, “The flooding time synchronization protocol,” in *Proc. of the 2nd International Conference on Embedded Networked Sensor Systems*, ACM, pp. 39-49, Nov. 2004. Article (CrossRef Link).

[18] M. Leng and Y.-C. Wu, “On clock synchronization algorithms for wireless sensor networks under unknown delay,” *IEEE Transactions on Vehicular Technology*, vol. 59, no.1, pp. 182-190, 2010. Article (CrossRef Link).

[19] L. Chaari and L. Kamoun, “Wireless sensors networks MAC protocols analysis,” *Journal of Telecommunications*, vol. 2, no. 1, pp. 42-48, Apr. 2010.

[20] M. A. Ameen, S. M. Riazul Islam, and Kyung Sup Kwak, “Energy Saving Mechanisms for MAC Protocols in Wireless Sensor Networks,” *International Journal of Distributed Sensor Networks*, vol. 2010, pp. 1-16, October 2010. Article (CrossRef Link).

[21] Amre El-Hoiydi and Jean-Dominique Decotignie, “WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks,” in *Proc. of Algorithmic Aspects of Wireless Sensor Networks*, Lecture Notes in Computer Science, vol. 3121, pp. 18-31, 2004. Article (CrossRef Link).

[22] Qualnet simulator, http://web.scalable-networks.com/content/qualnet.

[23] Ho Sy Khanh, Cheol-Young Ock and Myung Kyun Kim, “RP-MAC: A cross-layer duty cycle MAC protocol with a Reduced Pipelined-forwarding feature for Wireless Sensor Networks,” in *Proc. of IWCMC 2015*, pp. 1469-1474, Aug. 2015. Article (CrossRef Link).

Ngoc Minh Nguyen received a B.Sc. degree from Hanoi University of Science and Technology of Vietnam in 2012. He is now working and studying for M.S degree in School of IT Convergence of University of Ulsan in Korea. His main research area is MAC communication in wireless sensor network.

Myung Kyun Kim Ph.D., is currently a professor in School of IT Convergence, University of Ulsan, Korea. His main research area is real-time communications in wireless networks and industrial communication networks. He received the B.Sc. degree from Seoul National University of Korea and the M.Sc. and Ph.D. degrees from KAIST, Korea.