Performance Evaluation of the RIX-MAC Protocol for Wireless Sensor Networks

Taekon Kim[1] and Hyungkeun Lee[2]
[1] Department of Electronics and Information Engineering
Korea University
Sejong, Korea
[2] Department of Computer Engineering
Kwangwoon University
Seoul, Korea
[e-mail: taekonkim@korea.ac.kr , hklee@kw.ac.kr]
*Corresponding author: Hyungkeun Lee

Received October 9, 2016; revised December 8, 2016; accepted January 17, 2017; published February 28, 2017

Abstract

Energy efficiency is an essential requirement in designing a MAC protocol for wireless sensor networks (WSNs) using battery-operated sensor nodes. We proposed a new receiver-initiated MAC protocol, RIX-MAC, based on the X-MAX protocol with asynchronous duty cycles. In this paper, we analyzed the performance of RIX-MAC protocol in terms of throughput, delay, and energy consumption using the model. For modeling the protocol, we used the Markov chain model, derived the transmission and state probabilities, and obtained the equations to solve the performance of throughput, delay, and energy consumption. Our proposed model and analysis are validated by comparing numerical results obtained from the model, with simulation results using NS-2.

Keywords: wireless sensor network, medium access control, Markov chain model, asynchronous duty cycling, receiver-initiated protocol

The research has been conducted by the research grant of Kwangwoon University in 2014.

1. Introduction

The wireless sensor network (WSN) is an emerging technology that has a wide range of potential applications such as environmental monitoring, healthcare, military defense, and so on. It usually consists of a large number of densely deployed sensor nodes that are usually equipped with limited power and communicate with each other to cover the target area [1]. Since the replacement or recharge of batteries might be not possible, energy-efficiency is a fundamental and important consideration when designing sensor nodes in WSNs. Therefore, the energy constraint has been challenges in designing sensor nodes with energy-efficient hardware and protocols [2]. The protocols in WSNs include a MAC protocol which is an important technique to reduce the chance of collisions and enable the successful delivery of data frames. The key design goal of MAC protocols in WSNs is to provide energy efficiency, and others are throughput and latency to report the detected events to the sink promptly and sample the information with fine temporal resolution [3].

One of the most popular mechanisms to reduce energy consumption in WSNs is duty cycling, where a sensor node is periodically placed into the sleep mode [4]. With a lower duty cycle, sensor nodes can sleep longer and save more energy, whereas fewer sensor nodes participate in the delivery of data frames at any given time which will increase latency and decrease the throughput. Therefore, there is a trade-off between energy efficiency, latency, and throughput by the duty cycles. Existing duty cycling energy-efficient MAC protocols can be categorized into two types: synchronous and asynchronous protocols [3]. In synchronous MAC protocols such as S-MAC [5] and T-MAC [6], sensor nodes are synchronized to wake up at the same time by exchanging periodical control frames to obtain the delay reduction and throughput improvement. These additional control frames cause wasted energy, and a higher chance to access transmission medium at the same time causes collision between frames. On the other hand, asynchronous MAC protocols such as B-MAC [7], X-MAC [8], and PW-MAC [9] focus on the reduction of control frames to maintain the synchronization and efficient setup of communication channels among sensor nodes with different sleep schedules. Asynchronous MAC protocols are effective in energy saving because of the small number of control frames and low likelihood of collisions, but it is not easy to match wake-up states between communicating nodes. In particular, X-MAC is an energy-efficient MAC protocol using short preamble sampling for duty cycled WSNs, which is suitable for TinyOS, a widely-used operating system for sensor nodes [10].

We presented an energy-efficient receiver-initiated MAC protocol called RIX-MAC [11]. Although the operation of RIX-MAC is substantially similar to that of X-MAC, RIX utilizes the receiver-initiated wake-up scheme to reduce the number of short preambles of the sender and to avoid collisions partially by using random back-off scheme. As a result, RIX-MAC achieves higher performance with less energy than X-MAC and PW-MAC. We showed the performance evaluation results of RIX-MAC by the simulation compared with other energy-efficient MAC protocols. However, it is desirable to devise a general framework that can analyze the performance of X-MAC and RIX-MAC such as throughput, delay, and energy consumption.

In this paper, we propose a Markov chain model for duty-cycled nodes with a finite queue capacity in RIX-MAC, the analytical performance of RIX-MAC driven from the model and performance comparison with X-MAC. Furthermore, the simulation results are provided to validate the model and the analyses for RIX-MAC. The Markov chain model for nodes in S-MAC and X-MAC, and the performance analysis have been proposed [12]. In summary, the main contributions of this paper are 1) modeling and performance analyses for RIX-MAC by utilizing the Markov chain with a finite queue, 2) the validation of the model and analyses by comparing with the simulation results, and 3) the performance evaluation of RIX-MAC and X-MAC in both analysis and simulation.

The rest of this paper is organized as follows. Section 2 describes asymchronous MAC protocols and the operations of the X-MAC and RIX-MAC. Section 3 presents the Markov chain model for X-MAC and RIX-MAC and provides analytical relations of the network throughput, frame delay, and energy consumption using the model. Section 4 provides the numerical results compared with the simulation results to validate the model. Finally, concluding remarks will be given in section 5.

2. Operations of Asynchronous MAC Protocols

With a duty cycle, a sensor node spends most of the time in the sleep mode comsuming extremely low energy, and wakes up periodically to check whether there are frames destined for itself. In asynchronous MAC protocols, every node has its independent sleep schedule. Without overhead for synchronizing comunicating nodes' schedules, asynchronous MAC protocols can achieve low-duty cycle for energy efficiency, but they should find out efficient ways to establish communication between nodes.

To inform a receiver an impending data frame, a sender precedes the data frame with a preamble to be sampled by all potential receivers. This approach is suitable for systems with light traffic load where occasional data transmissions do not cause much overhead and channel contention. Because the preambles occupy the channel and prevent other nodes from transmission, the throughput achieved by preamble sampling is limited. For higher throughput and lower energy consumption, asynchronous MAC protocols are able to adopt receiver-initiated scheme to release medium for other nodes's transmission. In this section, we will review the operations of two asynchronous protocols, X-MAC and RIX-MAC, that use short preambles and receiver-initiated transmission.

2.1 X-MAC Using Asynchronous Duty-cycles

X-MAC [8] is an asynchronous duty-cycled MAC protocol for wireless sensor networks which is based on B-MAC [7] and uses a series of short preambles to achieve low-power communication without synchronization. Since the short preambles carry the address of the receiving node, other nodes rather than the receiving node can go to sleep as they hear the first short preamble. The receiving node can reply with an early-ACK to stop the preamble and start the data transfer. The key features of X-MAC are the short preamble sampling and the early-ACK mechanism. X-MAC transmits multiple short preambles to make the receiver notice instead of a long preamble used in B-MAC. If the receiver detects a short preamble during its wake-up period, it transmits an early-ACK frame to the sender before receiving the actual data frame. The sender knows that the receiver stays in the wake-up period, and it transmits data frame immediately. Consequently, X-MAC saves more energy than B-MAC due to the short preamble sampling and the early-ACK mechanism.

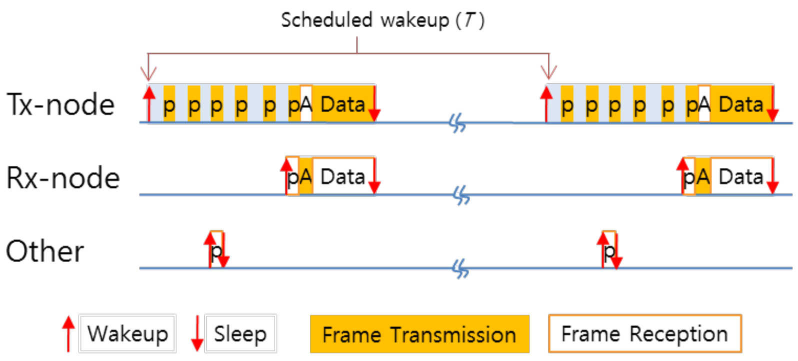


Fig. 1. Operations of X-MAC

Every node in the network has its own wake-up schedule to transmit and receive frames with a cycle of *T*, as shown in **Fig. 1**. Tx-node keeps sending short preambles until Rx-node wakes up and sends an early-acknowledgement back to Tx-node, while other nodes do not interfere with the communication between Tx-node and Rx-node. When Tx-node receives the acknowledgement, it starts transmitting a data frame to Rx-node.

Since X-MAC is asynchronous, the average communication time for Tx-node is *T*/2 plus the length of a data frame. When more than one Tx-node wake up and start sending their preambles at the same time, all the other nodes including Rx-node cannot determine address information in the preambles. Therefore, Tx-node will not stop sending preambles until their next wake-up shedule. Hence, for each colliding data frame, the average communication time for the sender is *T*.

2.2 RIX-MAC Using Receiver Initiation

RIX-MAC [11] is an extended protocol of X-MAC by employing a feature of receiver initiation. Every node in the network has the identical duty cycle of *T*, where each node alternates between wake-up and sleep states independently. A node is able to receive or transmit data frame through the radio in the wake-up state, and save the power by turning off the radio in the sleep state. In RIX-MAC, two types of wake-ups by schedule and synchronization cause a node to enter two wake-up states: the *Sched-wakeup* (scheduled wake-up) and *Synch-wakeup* (synchronized wake-up) states. A node enters the *Sched-wakeup* state periodically by its own wake-up schedule and the *Synch-wakeup* state optionally to transmit a data frame synchronized to the receiver.

The basic operations of RIX-MAC are shown in **Fig. 2**. Tx-node with a data frame destined to Rx-node, starts to transmit short preambles after the *Sched-wakeup*, and keeps transmitting short preambles until receiving the early-ACK. Rx-node wakes up at the *Sched-wakeup* and checks if there are any incoming short preambles intended for itself. If there are no incoming short preambles, Rx-node goes back to sleep. However, Rx-node replies with early-ACK to Tx-node if it finds out an incoming short preamble. Tx-node can obtain Rx-node’s wake-up schedule by the *wake-up time* field in early-ACK frames. Upon receiving an early-ACK from Rx-node, Tx-node starts to transmit a data frame immediately as well as records the *wake-up time* of Rx-node in the wake-up time table.
The first cycle of the transmission of RIX-MAC is the same as that of X-MAC since Tx-node has no information about Rx-node’s wake-up schedule at the beginning. Whenever it

transmits a data frame to Rx-node, Tx-node searches a wake-up table to obtain the wake-up time of the designated Rx-node. If it cannot find the appropriate wake-up schedule for the Rx-node in the table, Tx-node keeps transmitting short preambles from its *Sched-wakeup* till the reception of early-ACK.

When it has a next data frame to send to Rx-node, Tx-node wakes up again at the *Synch-wakeup* which is synchronized to Rx-node. When the channel stays idle, Tx-node puts a short preamble preceding a data frame. Therefore, Tx-node wakes up twice in a cycle when it has a data frame to be delivered to Rx-node (*Synch-wakeup*) and when it samples preambles at its own schedule to receive data frames (*Sched-wakeup*).

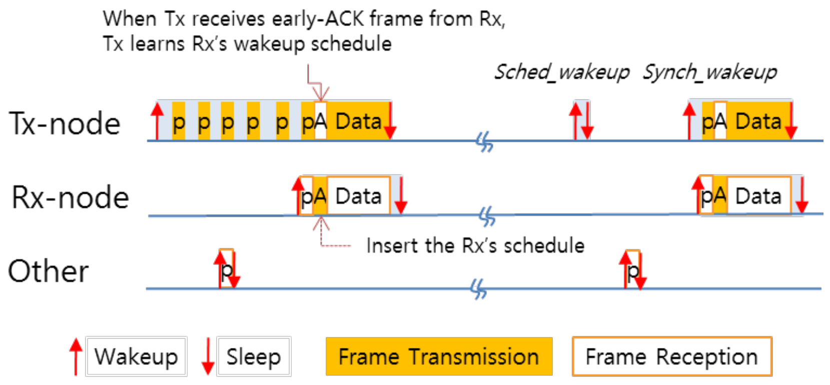


Fig. 2. Operations of RIX-MAC

Since Tx-node attempts to transmit data frames at the Rx-node's wake-up, RIX-MAC can induce a collision when multiple Tx-nodes try to deliver a data frame to an Rx-node. To coordinate the contention and avoid the collision of preambles in case of multiple Tx-nodes, RIX-MAC employs the random back-off scheme and the network allocation vector (NAV). An assigned random back-off for a Tx-node means the number of time slots to wait for the next transmission of short preamble after *Synch-wakeup*. An additional field of *duration* is inserted in preambles and early-ACK which contains information about slot-counts required to transmit a data frame for each TX-node. The other node may predict the time of the next data transmission. RX-node employs the *duration* field in early-ACK to coordinate transmissions of short preambles from contending multiple TX-nodes. TX-nodes acquiring the value of *duration* from TX-node's preamble or RX-node's early-ACK set NAV to defer from accessing the medium and stop to decrement slot-counts. After the completion of transmission, TX-nodes under the virtual carrier-sensing with NAV resume decrementing slot-counts and try to transmit short preambles.

Fig. 3 shows that Tx-node1 and Tx-node2 are senders trying to transmit data frames to a receiver, Rx-node, while Tx-node1 precedes Tx-node2 in occupying the medium. Tx-node1 grabs the medium to transmit a short preamble and Rx-node reply to the preamble with an early-ACK. Tx-node2 receives the early-ACK from Rx-node and sets the NAV to the end of data transmission. The duration in the early-ACK frame specifies the number of time slots Tx-node2 has to wait during the data transmission. As data transmission is completed, Tx-node2 counts down the residual slot-count again and transmits the preamble at the next time slot.

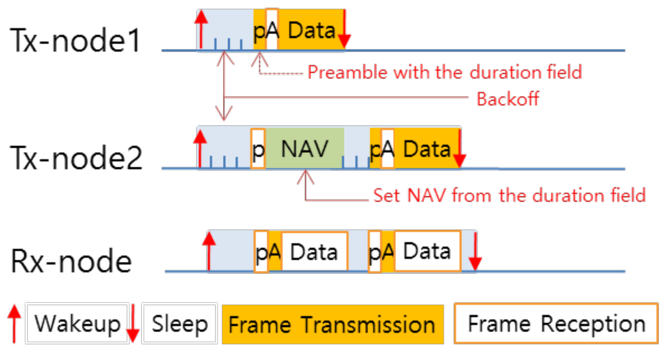


Fig. 3. Virtual carrier sensing and random back-off in RIX-MAC

3. Modeling of RIX-MAC

3.1 Overview

In order to model sensor nodes with a finite queue in duty-cycled sensor networks, we use a Markov chain with a finite number of states which represent queue length at the wake-up of nodes [12]. The Markov model assumes (1) independent frame arrivals at each node, (2) finite queue size at each node, (3) ideal channel (no fading, no capture effect, and no hidden terminals), (4) one transmission opportunity or one reception per cycle at each node, (5) no retransmissions, and (6) every node has a constant probability of transmitting a frame regardless of any node's queue length (similar assumptions were made in reference [13] and were verified as good approximations of real scenarios). This model is utilized to compute the throughput, the delay, and the energy consumption of RIX-MAC.

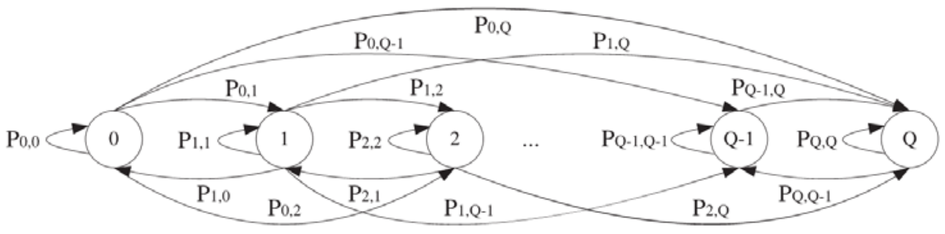


Fig. 4. Markov chain model of a node in the RIX-MAC protocol

Fig. 4 shows the Markov model with the queue capacity of Q, where each state represents the number of data frames in the queue. When the queue is not empty, a node will attempt to access the media to transmit a data frame. A data frame is removed from the queue when it is transmitted, and a data frame is dropped when the queue overflows. When A_i is the arrival probability of i data frames at a node and p is the probability of transmitting a frame in a cycle, the transition probabilities from one state to another are:

$$P_{0,i} = A_i,\ i = 0 \ldots Q-1\ , \tag{1}$$
$$P_{0,Q} = A_{\geq Q}\ , \tag{2}$$
$$P_{i,i-1} = p \cdot A_0,\ i = 0 \ldots Q\ , \tag{3}$$
$$P_{i,j} = p \cdot A_{j-i+1} + (1-p) \cdot A_{j-i}\ ,\ i = 1 \ldots Q-1,\ j = i \ldots Q-1\ , \tag{4}$$
$$P_{i,Q} = p \cdot A_{\geq Q-i+1} + (1-p) \cdot A_{\geq Q-i}\ ,\ i = 1 \ldots Q, \tag{5}$$
$$P_{i,j} = 0\ ,\ i = 2 \ldots Q,\ j = 0 \ldots i-2\ , , \tag{6}$$

In particular, Equations 2 and 5 show that frames are dropped when the queue overflows. When π_i is the stationary probability of a state i in the state space, $\tilde{S} = 0,1,\ldots,Q$, a stationary distribution $\pi = (\pi_0,\ldots,\pi_Q)$ exists since the Markov model is irreducible and aperiodic. For any $\mathrm{s}_i \in \tilde{S}$:

$$\textstyle\sum_{s_i \in \tilde{S}} \pi_i = 1, \text{ and} \qquad \pi\tilde{P} = \pi \tag{7}$$

Here, $\tilde{P}$ is the transition matrix. If frame arrival information is known, the probability p for each node to win the contention becomes the only variable in the transition matrix, $\tilde{P}$ and π_i can be represented as a function of p. Therefore, the relation between π_0 and p for each node can be described using a function, $f(\cdot)$, by:

$$\pi_0 = f(p). \tag{8}$$

It tells that the stationary probability, π_0 , of the empty-queue state can be obtained using the proposed Markov model for a given probability p for each node to win the contention. In addition to Equation 8, another relationship between π_0 and p is required to solve both π_0 and p, which can be obtained from the media access rules of a MAC protocol.

3.2 Average Throughput and Delay

RIX-MAC is an asynchronous time-slotted protocol, where T is the length of a cycle, T_{ACT} is the active period of a cycle, and τ is the time slot. In RIX-MAC, every node wakes up periodically at its own schedule (*Sched-wakeup*) and wakes up to send a data frame synchronized to the receiver's wake-up (*Synch-wakeup*). RIX-MAC may have collisions when more than one transmitter wakes up and starts sending their preambles after the same back-off.

If there are N nodes in a fully connected network, when a node has a data frame to send, the probability $M_{j,k}$ that k nodes out of j nodes wake up synchronizing to a receiver (*Synch-wakeup*) when j nodes have frames to transmit in the queues competing for transmitting a frame to the receiver can be described as a function of π_0 by:

$$M_{j,k} = \binom{N-1}{j}(1-\pi_0)^j {\pi_0}^{N-1-j} \cdot \binom{j}{k}\left(\frac{1}{N-1}\right)^k \left(\frac{N-2}{N-1}\right)^{j-k} ,j = 0 \ldots N-1,\ k = 0 \ldots j \quad (9)$$

In the case that k nodes wake up synchronized to the receiver and compete for the channel by transmitting a preamble frame. The probability p_k of successfully delivering the preamble frame to the receiver can be calculated by:

$$p_k = \sum_{i=1}^{W} \frac{1}{W}\left(\frac{W-i+1}{W}\right)^k,\ k = 0 \ldots N-1\ , \quad (10)$$

Here, W is the contention window size. Therefore, the probability p for each node to get the channel and the probability p_s for each node to successfully transmit the data frame can be derived as follows:

$$\begin{aligned} p &= \sum_{j=0}^{N-1} \sum_{k=0}^{j} M_{j,k} \cdot p_k \\ &= \sum_{j=0}^{N-1} \sum_{k=0}^{j} \binom{N-1}{j}(1-\pi_0)^j {\pi_0}^{N-1-j} \cdot \binom{j}{k}\left(\frac{1}{N-1}\right)^k \left(\frac{N-2}{N-1}\right)^{j-k} \sum_{i=1}^{W} \frac{1}{W}\left(\frac{W+1-i}{W}\right)^k \quad (11) \end{aligned}$$

$$p_s = \sum_{j=0}^{N-1} \sum_{k=0}^{j} \binom{N-1}{j}(1-\pi_0)^j {\pi_0}^{N-1-j} \cdot \binom{j}{k}\left(\frac{1}{N-1}\right)^k \left(\frac{N-2}{N-1}\right)^{j-k} \sum_{i=1}^{W} \frac{1}{W}\left(\frac{W-i}{W}\right)^k \quad (12)$$

With Equations 11 and 7, stationary probability π_0 can be obtained and plugging π_0 into (12), the probability for each node to successfully transmit a data frame, p_s, can be determined. Exploiting π_0, p, and p_s, the average throughput and delay in the network can be obtained. The throughput is defined as the amount of data successfully delivered within the time elapsed. Since the RIX-MAC protocol operates with duty cycles, the throughput can be calculated within a cycle time. Therefore, the average throughput of RIX-MAC is:

$$\text{Throughput} = \frac{Total\ amount\ of\ data\ delivered}{Total\ time\ elapsed} = \frac{N\ (1-\pi_0)\, p_s \cdot S}{T}, \tag{13}$$

Here, S is the size of a frame. The delay experienced by a frame consists of two components such as a queueing delay, D_Q, and a contending delay, D_C, resulting in $D = D_Q + D_C$. The queuing delay is the time that a frame must stay in the queue until all preceding frames are transmitted. The contending delay is defined as the interval from the time of a frame entering the queue to the time of the frame transmitted from the queue as shown in [12]. During a cycle of T,

$$D_C = T \cdot \sum_{i=0}^{\infty} (\, i+1) p (1-p)^i \text{ , and} \tag{14}$$
$$D_Q = D_C \cdot \left\{ \sum_{i=1}^{Q-1} (i - 0.5) \cdot \frac{\pi_i}{1-\pi_Q} \right\}. \tag{15}$$

By the Markov model, the queuing delay, D_Q, for a data frame can be obtained by a function of the contending delay, D_C, of preceding data frames. A data frame may enter the queue when the frame at the head of the queue is in the middle of contending process for transmission media and has to wait on average for a half of the contending delay. To obtain the contending delay, D_C, a node with a data frame contends for the media once in a cycle of T, where the node has a probability of p to win the contention. Therefore, the total delay for a data frame is calculated as the sum of two delay components, (14) and (15).

3.3 Energy Consumption

Since RIX-MAC is a duty-cycled protocol, the average energy consumption over a cycle, P, is obtained by dividing the energy consumption, E, of a node in a cycle with the cycle length, T. RIX-MAC protocol has two active periods, *Sched-wakeup* and *Synch-wakeup*, where the *Synch-wakeup* period is optional when a node wants to transmit data frames. During a periodic wake-up (*Sched-wakeup*), Rx-node can successfully receive a data frame if only one node in the network has a data frame destined to the Rx-node or more than two nodes have data frames destined to the Rx-node with different back-off values. Also, Rx-node fails to receive a data frame if more than two nodes have data frames destined to the Rx-node picking up an identical back-off value. Therefore, in Tx-node at *Synch-wakeup*, the node performs one of the following operations:

1) Successful transmission of a data frame with the probability of $(1-\pi_0)p_S$,
2) Collided transmission of a data frame with the probability of $(1-\pi_0)(p-p_S)$, or
3) Sleeping without any transmission with the probability of $\pi_0+(1-\pi_0)(1-p)$.

An Rx-node at *Sched-wakeup* operates in one of the following:

1) Successful reception of a data frame with the probability of $(N-1)(1-\pi_0)\pi_0{}^{N-2}+\sum_{j=2}^{N-1}\sum_{k=1}^{j}M_{j,k}\cdot p_k=p_{RS}$,
2) Collision of data frames with the probability of $\sum_{j=2}^{N-1}\sum_{k=2}^{j}M_{j,k}(1-p_k)=p_{RF}$, or
3) Staying active and idle with the probability of $\pi_0{}^{N-1}+\sum_{j=1}^{N-1}M_{j,0}=p_I$.

In order to calculate the energy consumption, we have duration periods of a preamble, a data frame, and an early-ACK frame denoted by t_{Pre}, t_{Data} , and t_{Ack}, respectively. We also have the power for a node to transmit a frame, receive a frame, and sleep denoted by P_{Tx}, P_{Rx}, and P_S, respectively.

Therefore, we can obtain the average energy consumption in a node during a cycle by calculating it for each case. First, we will derive the energy consumption of Tx-node during a *Synch-wakeup* period. It takes the time of $\frac{W}{2}+t_{Pre}+t_{Ack}+t_{Data}$ on average to transmit a data frame, where $\frac{W}{2}$ is the average duration for random back-off. The energy consumption that a node transmits a data frame successfully at *Synch-wakeup*, E_{TS}, is:

$$E_{TS}=\tau\left(t_{Pre}\cdot P_{Tx}+t_{Ack}\cdot P_{Rx}+t_{Data}\cdot P_{Tx}+\frac{W}{2}\cdot P_{Rx}\right). \tag{16}$$

When preambles collide at *Synch-wakeup*, no preamble frame can be received and Tx-nodes keep transmitting preambles during residual active period. Therefore, the energy consumption that nodes transmit a data frame and encounter a collision at *Synch-wakeup*, E_{TF}, is:

$$E_{TF}=\tau\left(\frac{W}{2}\cdot P_{Rx}\right)\left\{\left(T_{ACT}-\frac{W}{2}\right)\left(\frac{t_{Pre}}{t_{Pre}+t_{Ack}}\cdot P_{Tx}\right)\left(\frac{t_{Ack}}{t_{Pre}+t_{Ack}}\cdot P_{Rx}\right)\right\}, \tag{17}$$

The energy consumption of a sleeping node is assumed to be negligible such that $E_S \approx 0$. During a *Sched-wakeup* period, the node occupying the channel transmits a preamble, receives an acknowledgement, and sends a data frame just after a random back-off, which is $t_{Pre} + t_{Ack} + t_{Data}$. When a node receives a data frame successfully at *Sched-wakeup*, the energy consumption, E_{RS}, is:

$$E_{RS} = \tau\left\{(t_{Pre} \cdot P_{Rx} + t_{Ack} \cdot P_{Tx} + t_{Data} \cdot P_{Rx}) + (T_{ACT} - (t_{Pre} + t_{Ack} + t_{Data} + \frac{W}{2}))P_{Rx}\right\}. \quad (18)$$

Collision occurs when two or more Tx-nodes have the same receiver and choose the same contention window size at *Sched-wakeup*. The energy consumption that a node receives data unsuccessfully in *Sched-wakeup* period, E_{RF}, is:

$$E_{RF} = \tau\left(t_{Pre} + \frac{W}{2}\right) P_{Rx} . \quad (19)$$

Finally, a node may stay active and idle during the period of T_{ACT}, neither sending nor receiving a data frame. The energy consumption of a node staying idle at *Sched-wakeup*, E_I, is:

$$E_I = \tau \cdot P_{Rx} \cdot T_{ACT} . \quad (20)$$

Table 1. Cases of nodes' status in a cycle

Sched_wakeup / ***Synch_wakeup***	Rx_Success	Rx_Failure	Idle
Tx_Success	Case 1	Case 2	Case 3
Tx_Failure	Case 4	Case 5	Case 6
Sleep	Case 7	Case 8	Case 9

Since every node in RIX-MAC has two independent periods, to transmit and receive at *Sched-wakeup*, the operations of a node in a cycle can be classified into nine cases, which could occur in a cycle as shown in **Table 1**. Case 1 implies that a node successfully transmits a data frame and receives a data frame during the *Synch-wakeup* and *Sched-wakeup* periods, respectively. Case 2 shows that a node successfully transmits a data frame but fails to receive a data frame at *Sched-wakeup*, while Case 3 shows that a node only transmits a data frame successfully at *Synch-wakeup* and no data frame is destined to the node. Similarly, other cases represent the operations of a node in a similar manner. Therefore, the average energy consumption of a node in a cycle can be obtained by the sum of the energy multiplied by the probability of every case as shown in **Table 2**.

Table 2. Energy consumption of cases in **Table 1**

Cases	**Energy consumption**
1	$(1-\pi_0)p_s \cdot p_{RS}(E_{TS}+E_{RS})$
2	$(1-\pi_0)p_s \cdot p_{RF}(E_{TS}+E_{RF})$
3	$(1-\pi_0)p_s \cdot p_I(E_{TS}+P_{Rx} \cdot T_{ACT})$
4	$(1-\pi_0)(p-p_s)p_{RS}(E_{TF}+E_{RS})$
5	$(1-\pi_0)(p-p_s)p_{RF}\ (E_{TF}+E_{RF})$
6	$(1-\pi_0)(p-p_s)p_I\ (E_{TF}+P_{Rx} \cdot T_{ACT})$
7	$(1-p+p \cdot \pi_0)p_{RS} \cdot E_{RS}$
8	$(1-p+p \cdot \pi_0)p_{RF} \cdot E_{RF}$
9	$(1-p+p \cdot \pi_0)p_I \cdot P_{Rx} \cdot T_{ACT}$

4. Performance Analysis

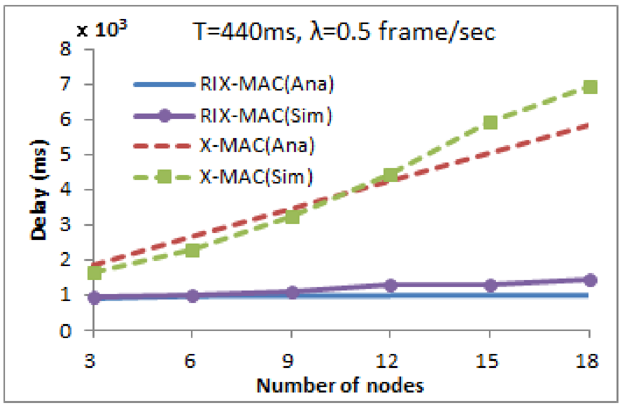
In this section, we present the performance evaluation of RIX-MAC in terms of the throughput, delay, and energy consumption of the network. In order to validate our proposed model of RIX-MAC and evaluate the performance of RIX-MAC, we compare the numerical results of the proposed model with the simulation results using NS-2 [14]. We also show the performance of RIX-MAC against the performance of X-MAC in terms of delay, throughput, and energy consumption. For the analysis and simulation, it is assumed that the network is fully connected and each node has a single omni-directional antenna where the radio propagation model of two-way ground reflection is employed. A queue capacity, Q, at each node is set to 10. A slot time, τ, is 1ms and the active time in a cycle, T_{ACT}, is 40 slots. The time durations to transmit a preamble frame, t_{Pre}, and early-ACK frame, t_{Ack}, are set to one slot and the time used to transmit a data frame, t_{Data}, is 5 slots. Other key parameters of nodes and networks are shown in **Table 3**.

Table 3. Parameters for sensor networks

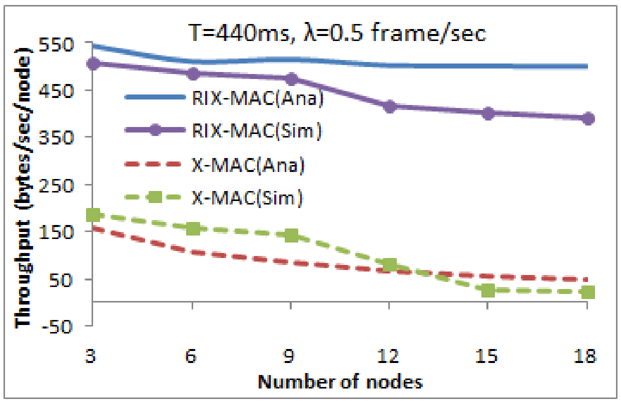
Parameters	**Notation**	**Value**
Channel bandwidth	BW_{ch}	250 kbps
Active period in a cycle	T_{ACT}	40 ms
Time for a preamble frame	t_{Pre}	1 ms
Time for an ack frame	t_{Ack}	1 ms
Time for a data frame	t_{Data}	5 ms
Power in Rx	P_{Rx}	15.2 mW
Power in Tx	P_{Tx}	28.9 mW
Power in sleep state	P_S	0.0004 mW ≈ 0 mW
Queue length	Q	10
Back-off window	W	32
Total simulation time	T_{SIM}	1,000 sec

The stationary probability of the empty-queue state, π_0, and the probability for each node to win the contention, p, can be obtained by solving Equations 7, 8, and 11, whereas the stationary distribution of the model, π, can be calculated by plugging p into Equations 1 through 6. As described in Equation 11, a pair of probabilities, (p, π_0), under various duty-cycle ratios and parameters of **Table 3**, is used to obtain the throughput using Equation 13. For example, we obtain p to be 0.77 and π_0 to be 0.42 in the case where the duty cycle is 14.3%, 12 nodes reside in the network, and the back-off window is equal to 32.

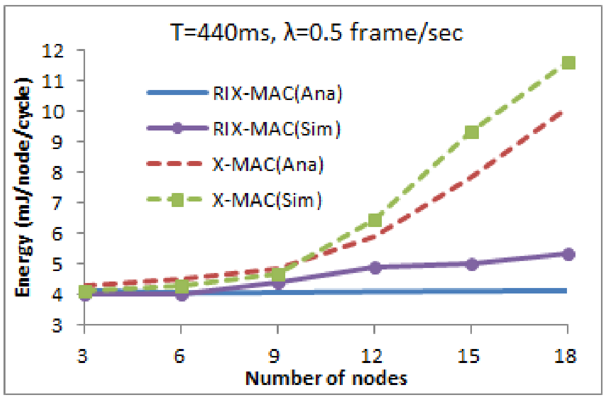
Fig. 5 through **Fig. 7** show the results of performance from both the model and the simulation in both RIX-MAC and X-MAC under various environments, where the dashed lines indicate performance results of X-MAC and the solid lines represent the performance results of RIX-MAC. Fig.s in (a) show the average throughput, Fig.s in (b) show the average delay, and Fig.s in (c) show the total energy consumption under varying data arrival rate, cycle length, and number of nodes in the network.



(a) Average delay vs. Number of nodes

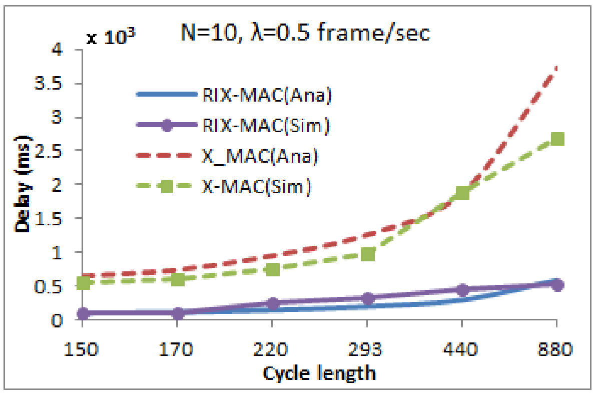


(b) Average throughput vs. Number of nodes

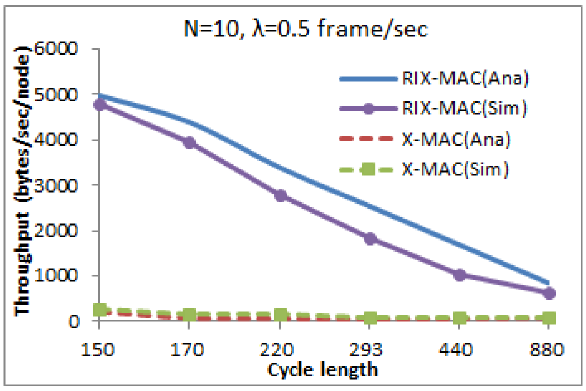


(c) Energy consumption vs. Number of nodes

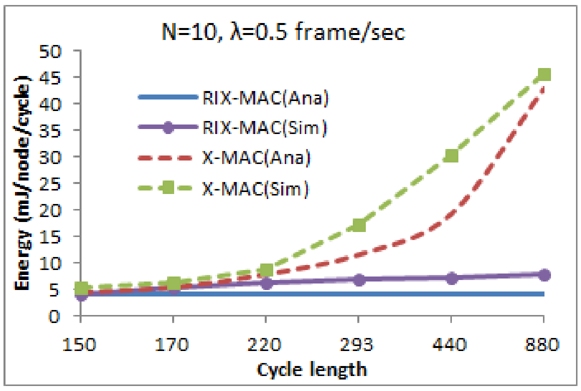
Fig. 5. Performance of X-MAC and RIX-MAC vs. Number of nodes



(a) Average delay vs. Cycle length

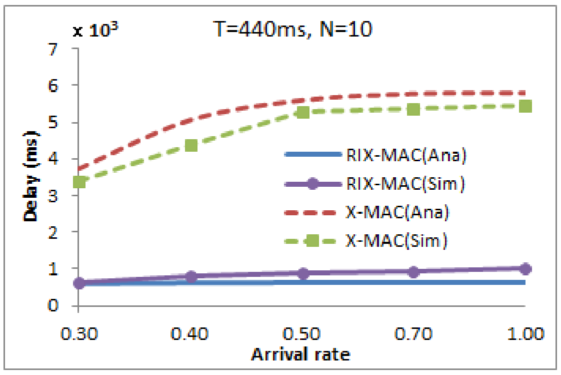


(b) Average throughput vs. Cycle length

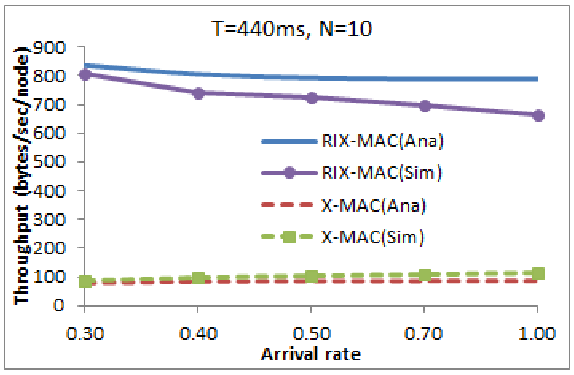


(c) Energy consumption vs. Cycle length

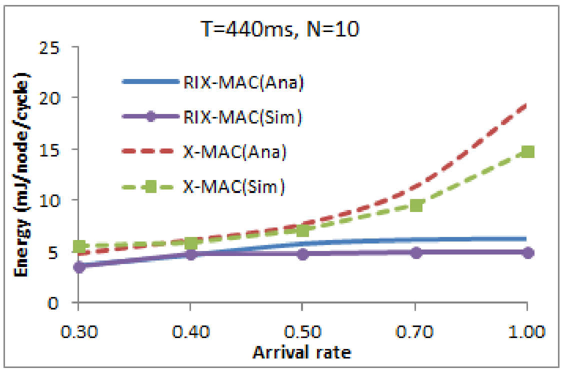
Fig. 6. Performance of X-MAC and RIX-MAC vs. Cycle lengths



(a) Average delay vs. Arrival rate



(b) Average throughput vs. Arrival rate



(c) Energy consumption vs. Arrival rate

Fig. 7. Performance of X-MAC and RIX-MAC vs. Arrival rates

In **Fig. 5(a)**, the delay of X-MAC increases linearly as the number of nodes increases, while the delay of RIX-MAC increases slightly. When there are more nodes in the network using X-MAC, the chance for each node to be able to transmit a frame in a cycle becomes lower, and the probability of a longer queue gets higher to yield a longer delay. In **Fig. 5(b)**, the throughput per node of both X-MAC and RIX-MAC is not influenced by the number of nodes in the network because both successful transmissions increase along with failures of transmission (collisions) as the number of nodes increases. However, RIX-MAC outperforms X-MAC in terms of throughput per node. The energy consumption per node during a cycle increases in X-MAC as the number of nodes in the network increases, as shown in **Fig. 5(c)**, since X-MAC wastes energy to transmit short preambles even in the case of collisions. However, since nodes in RIX-MAC may fail to access the channel and goes to sleep without consuming energy to transmit preambles, RIX-MAC is able to save energy.

When cycle length is small, all the incoming frames can be transmitted with waiting less after they arrive in the network as shown in **Fig. 6(a)**. However, as the cycle length increases, the contention delay and the queuing delay increase because of saturations and queue overflows. In **Fig. 6(b)**, the throughput of RIX-MAC changes little as cycle lengths increase. However, the throughput of X-MAC decreases linearly as the cycle length increases because of low duty cycles and short active periods to deliver incoming frames. A longer cycle length leads to lower throughput since it is assumed that, at most, one frame in a cycle can be delivered. In **Fig. 6(c)**, the energy consumption per node during a cycle maintains a constant in RIX-MAC regardless of the cycle length because all the nodes that are not involved in the delivery of frames may sleep for a longer time.

In X-MAC, the delay of RIX-MAC first increases almost linearly as the arrival rate increases during relatively low arrival rates, but it does not increase linearly with higher arrival rates because of overflows in the queue and the contention in the network, as shown in **Fig. 7(a)**. In **Fig. 7(b)**, the delay of RIX-MAC does not increase as the arrival rate increases because the queue at each node is not filled with frames with these arrival rates. In **Fig. 7(c)**, energy consumption per node increases when the load of incoming frames increases because more energy is required to deliver more frames. However, when RIX-MAC saturates and the queue at each node overflows, more frames are dropped by the queue and the number of frames to be transmitted remains the same. Therefore, the energy consumption of RIX-MAC becomes limited as the load of incoming frames increases further.

5. Conclusion

We have proposed the modeling and analysis of RIX-MAC, a new energy-efficient MAC protocol for sensor networks based on asynchronous duty cycling. We have also presented the performance evaluation of RIX-MAC, comparing the performance of RIX-MAC with the performance of X-MAC. Energy saving has been a research issue for wireless sensor networks since the lifetime of networks is critical. RIX-MAC is designed to minimize energy consumption by using short preambles and enabling senders to predict a receiver's wake-ups. Transmission collisions may occur over wireless channels, especially with bursty traffic that may be usual in a sensor network. The backoff mechanism of RIX-MAC also reduces the likelihood of transmission collisions in the case of multiple senders. We have conducted the simulation using the NS-2 simulator to evaluate the performance of RIX-MAC and X-MAC. Simulation results show that the RIX-MAC protocol outperforms the X-MAC protocol in terms of delay, throughput, and energy consumption by avoiding collision and reducing overhead.

As a result, RIX-MAC can achieve speedy communication in wireless sensor networks without sacrificing energy saving, and it can increase the lifetime of sensor networks by lower collision and unnecessary wake-ups. However, these superior properties of RIX-MAC depends on many operational factors. It needs to be implemented and evaluated in a wireless sensor network with mobile sensor nodes, since the scheme of receiver initiation can fall under the influence of mobility. The routing protocol and upper layers have not been taken into account in this research, which cooperate closely with a MAC protocol in practice. The effects of node densities need to be investigated since the large number of nodes tends to lower the performance of RIX-MAC. Therefore, more work needs to be done in future to obtain a valid evaluation for RIX-MAC with all the related factors taken into consideration.

References

[1] J. Yick, B. Mukherjee and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, Vol. 52, No. 12, pp. 2292-2330, August 2008. Article (CrossRef Link)

[2] S. Sendra, J. Lloret, M. García, J. F. Toledo, "Power saving and energy optimization techniques for wireless sensor networks," *Journal of communications*, Vol. 6, No. 6, pp. 439-459, September 2011. Article (CrossRef Link)

[3] P. Huang, L. Xiao, S. Soltani, M. W. Mutka and N. Xi, "The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, Vol. 15, No. 1, pp. 101-120, January 2013. Article (CrossRef Link)

[4] K. Han, J. Luo, Y. Liu and A. V. Vasilakos, "Algorithm Design for Data Communications in Duty-Cycled Wireless Sensor Networks: A Survey," *IEEE Communications Magazine*, Vol. 51, No. 7, pp. 107-113, July 2013. Article (CrossRef Link)

[5] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. of 21st International Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1567-1576, June 2002. Article (CrossRef Link)

[6] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proc. of the first ACM Conference on Embedded Networked Sensor System*, pp. 171-180, November 2003. Article (CrossRef Link)

[7] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. of the second ACM Conference on Embedded Networked Sensor System*, pp. 95-107, November 2004. Article (CrossRef Link)

[8] M. Buettner, G. V. Yee, E. Anderson and R. Han, "X-MAC: a short preamble mac protocol for duty-cycled wireless sensor networks," in *Proc. of the 4th ACM Conference on Embedded Networked Sensor System*, pp. 307-320, November 2006. Article (CrossRef Link)

[9] Lei Tang, Yanjun Sun, Omer Gurewitz, and David B. Johnson, "PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks," in *Proc. of the 30th IEEE International Conference on Computer Communications*, pp. 1305-1313, April 2011. Article (CrossRef Link)

[10] M. Amjad, M. Sharif, M. K. Afzal, S. W. Kim, "TinyOS-new trends comparative views and supported sensing applications: A review," *IEEE Sensors Journal*, Vol. 16, No. 9, pp. 2865-2889, May 2016. Article (CrossRef Link)

[11] Inhye Park, Hyungkeun Lee, and Seokjoong Kang, "RIX-MAC: An Energy-efficient Receiver-initiated Wakeup MAC protocol for WSNs," *International Journal of KSII Transactions on Internet and Information System*, Vol. 8, No. 5, May 2014. Article (CrossRef Link)

[12] Ou Yang, Wendi B. Heinzelman, "Modeling and Performance Analysis for Duty-Cycled MAC Protocols with Applications to S-MAC and X-MAC," *International Journal of IEEE Transactions on Mobile Computing*, Vol. 11, No. 6, pp. 905-921, June 2012. Article (CrossRef Link)

[13] K. Ghaboosi, B. H. Khalaj, Y. Xiao, and M. Latva-aho, "Modeling IEEE 802.11 DCF using parallel space-time Markov chain," *IEEE Transactions on Vehicular Technology*, Vol. 57, No. 4, pp. 2404-2413, July 2008. Article (CrossRef Link)

[14] NS-2, URL: http://www.isi.edu/nsnam/ns/

Taekon Kim received the Ph.D. degree in electrical engineering from the Pennsylvania State University, University Park, PA, in 2001. From 2001 to 2002, he was with Technology & Research Labs, Intel Corporation, Chandler, AZ. From 2003 to 2004, he was with Digital Media R&D center, Samsung Electronics, Suwon, Korea. Since 2005, he has been with Korea University, Korea, where he is an associate professor of Electronics and Information Engineering. His research interests include multimedia signal processing, wireless networks and communications.

Hyungkeun Lee received the B.S. degree in electronic engineering from Yonsei university, Seoul, Korea, in 1987, and M.S. and Ph.D. degree in computer engineering from Syracuse university, NY, USA in 1998 and 2002, respectively. He is an associate professor in the department of computer engineering at Kwangwoon university, Seoul, Korea. He was with Samsung Electronics as a senior research engineer for six years. His current research interests are in the area of wireless networks such as mobile ad hoc networks, wireless LAN, wireless sensor networks and wireless multimedia communication. He is also interested in design and analysis of tactical data links in defense systems.