Performance Evaluation of X-MAC/BEB Protocol for Wireless Sensor Networks

Ayaz Ullah and Jong-Suk Ahn

Abstract: **This paper proposes an X-MAC/BEB protocol that runs a binary exponential backoff (BEB) algorithm on top of an X-MAC protocol to save more energy by reducing collision, especially in densely populated wireless sensor networks (WSNs). X-MAC, a lightweight asynchronous duty cycle medium access control (MAC) protocol, was introduced for spending less energy than its predecessor, B-MAC. One of X-MAC 's conspicuous technique is a mechanism to allow senders to promptly send their data when their receivers wake up. X-MAC, however, has no mechanism to deal with sudden traffic fluctuations that often occur whenever closely located nodes simultaneously diffuse their sense data.**

To precisely evaluate the impact of the BEB algorithm on X-MAC, this paper builds an analytical model of X-MAC/BEB that integrates the BEB model with the X-MAC model. The analytical and simulation results confirmed that X-MAC/BEB outperformed X-MAC in terms of throughput, delay, and energy consumption, especially in congested WSNs.

Index Terms: **Binary exponential backoff (BEB) algorithm, duty cycle medium access control (MAC) protocol, wireless sensor networks, X-MAC protocol.**

I. INTRODUCTION

DUTY cycle medium access control (MAC) protocols have been developed mainly for sparing energy in wireless sensor networks (WSNs). WSNs consist of many nodes equipped with a transceiver and multiple sensors to cooperatively monitor environmental changes and if any, pass their sensed data to their central hub either directly or through a sequence of intermediate nodes. Since events to inform rarely tend to appear, only for a short period time duty cycle MAC protocols periodically wake up to either send or receive data. Except for their short wakeup time, they sleep most of the time, by turning off their transceiver to avoid idle listen to operations. Concisely a duty cycle alternates a lengthy sleep period and a brief wakeup period to spend as little energy as possible.

Duty cycle MAC protocols are divided into synchronous and asynchronous classes depending on whether or not a node wakes up at the same time as with its neighbors, this reduces idle time wasted until its receivers become awake. At the expense of the short waiting time, synchronous MAC protocols periodically exchange their schedule information among neighbor nodes. They have been evolved from S-MAC [1] to its variants such as T-MAC [2], RMAC [3], DW-MAC [4], and SCP-MAC [5] with an aim to improve overall energy efficiency.

For better energy efficiency, asynchronous duty-cycle MAC protocols such as B-MAC [6], WiseMAC [7], and X-MAC [8] take a different approach. Here nodes are permitted to function independently from their neighbors. For the freedom that relieves them of periodic synchronization chores, however, asynchronous senders need to keep transferring a preamble until their receiver wakes up. Once the presence of the preambles is detected, the receiver needs to be continuously awake to notify its awareness at the end of the non-preemptive preamble. Even though asynchronous MAC protocols tend to perform better, especially in WSNs conveying sporadic traffic, they suffer from these monolithic preambles lasting as long as one duty cycle. As WSNs hibernate for a longer time, they would consume more energy since longer sleep periods require longer preambles.

X-MAC, an enhanced version of B-MAC, has differentiated itself from its predecessors by incorporating two sophisticated features. First, X-MAC permits receivers to inform immediately of their availability, even in the middle of the preamble transmission, to lessen the waiting time of the senders. To make a long preamble interruptible, it delivers one short preamble after another with a gap during which an awaken receiver can instantly reply with an early acknowledgment (ACK). Second, X-MAC embeds the target address into each short preamble to let other overhearing nodes go back to sleep if they are not the intended receiver.

X-MAC successfully diminishes the duration of the preamble, it still degrades the performance under congested WSNs. X-MAC has no mechanism to deal with collision. Since X-MAC cannot recognize the occurrence of the collision, it wastes one whole cycle once a collision happens. Moreover, collisions are likely to occur at the retransmission time since the waiting time before the preamble delivery is statically randomized by the fixed contention window (CW) regardless of the number of collisions. Note that the binary exponential backoff (BEB) algorithm keeps doubling its CW as collisions consecutively occur.

As WSNs are overcrowded with nodes, collision is highly likely to happen due to a *spatially correlated contention* [9] phenomenon where several close-by nodes sensing the same event would simultaneously initiate the transfer of their sensor data. Without careful design and control, this phenomenon can cause overwhelming collisions, reducing throughput and energy efficiency. To address this problem, this paper proposes an X-MAC/BEB protocol, which augments X-MAC with a BEB algorithm to spread out the transmission time more widely in proportion to the degree of congestion. As WSNs become congested, the BEB scheme alleviates the impact of collisions by

Manuscript received March 18, 2015; approved for publication by Ali Abedi, Division III Editor, May 29, 2016.

This work has been supported by the research program of Dongguk University, 2013

Ayaz Ullah and Jong-Suk Ahn are with the Department of Computer Science and Engineering, Dongguk University, 26, Pil-dong 3-ga, Jung-gu, Seoul, Korea, email: {ayazsb, jahn}@dongguk.edu.

Digital object identifier 10.1109/JCN.2016.000114

dynamically randomizing the transmission time. This paper also introduces an analytical model of X-MAC/BEB by including the behavior of the BEB algorithm into a traditional X-MAC model [10]. The analytical and simulation results from the network simulator version 2 (ns-2) [11], verified with 3% discrepancy that the BEB mechanism boosts the performance of X-MAC in terms of throughput, delay, and energy usage by 40%, 15%, and 40%, respectively in a WSN with 40 nodes.

The rest of the paper is organized as follows: Section II overviews related work and Section III elaborates the X-MAC/BEB protocol in detail. Section IV explains a traditional Markov model of X-MAC protocol. Section V presents the extended performance model of X-MAC/BEB and Section VI compares the performance of X-MAC/BEB with that of X-MAC. Section VII concludes the paper and presents future work.

II. RELATED WORK

In terms of better performance, there have been modified versions of X-MAC such as LA-MAC [12], CL-MAC [13], EX-MAC [14], (VT) [15], RIX-MAC [16], and TRIX-MAC [17]. For speedy transmission, LA-MAC schedules senders to wake up in the ascending order of their corresponding receivers' wakeup time to shorten the senders' waiting time. It also minimizes the delivery delay by enabling senders to transmit multiple frames back-to-back to their receiver once the channel is captured. Note that the legacy X-MAC permits only two senders, at maximum, to deliver their frames in a row when they are destined to the same receiver. LA-MAC, however, performs poorly under the congested WSNs since it is not equipped with a mechanism to monitor collision and avoid a further collision like the X-MAC.

For lowering the end-to-end delay over multi-hop WSNs, CL-MAC incorporates a synchronous feature by containing the destination address and rendezvous (RDV) point in the preamble. The RDV point indicates the time at which all the neighbors around the sender should simultaneously wake up. After reading the preamble, all the neighbor nodes, including the receiver, adjust their next wakeup time according to the RDV point. To alleviate collision at the RDV point, when all active senders are likely to transmit, CL-MAC adopts a random backoff algorithm with fixed CW. CL-MAC, however, does not provide a complete analysis of the effect of the random backoff algorithm on throughput, delay, and energy expenditure.

Just like CL-MAC, EX-MAC intends to decrease the end-to-end delay to minimize energy consumption in multi-hop WSNs via a different technique. It delivers reservation requests in advance for synchronizing intermediate nodes between a source and its destination. EX-MAC does not employ a specific mechanism to address the congestion problem so that it may incur a longer delay, especially in congested WSNs due to frequent collision and extra reservation traffic.

As with the previous schemes, virtual tunnel (VT) also tries to fulfill the same goal of reducing the end-to-end delay in multi-hop WSNs by approximately synchronizing the duty cycle on demand by implementing synchronization at the neighbor's duty cycle (SND) mechanism. By inserting the schedule information in short preambles, SND enables each node to know the wakeup time of its neighbor nodes. As with EX-MAC, VT does not explicitly mention the collision problem that would be more severe with this on-demand synchronization.

Receiver-initiated X-MAC (RIX-MAC) and tree topology receiver-initiated X-MAC (TRIX-MAC) work in the same way in that senders recognize the wakeup time of their receiver by early ACK. These algorithms explicitly inform the exact wakeup time of receivers by loading the wakeup time into early ACK. This explicit notification, however, requires them to synchronize periodically the local clock of sensor nodes. This would be ineffective in WSNs with a large number of sensor nodes. Note that they do not clearly address this clock drift problem. They, furthermore, only adopts a static CW, not the BEB algorithm, which would be inadequate in dense WSNs.

With regard to performance models, Buettner *et al*. [8] analytically abstracts the energy consumption and latency of X-MAC and compares analytical results with measurements from real WSNs for validating the correctness of their model. This model, however, is limited to analyzing a simple network, with a pair of one sender and its receiver. It, furthermore, does not model the queueing behavior of duty-cycle protocols that data frames generated during sleep periods are queued in the buffer, leading to an inadequate performance prediction.

To solve this problem, Heinzelman *et al*. [10] proposes a performance framework combining two models, one for the duty-cycle behavior and another for a specific MAC protocol such as S-MAC and X-MAC. The first Markov model depicts transitions among the states of a node. Here, the state of a node is described as its current queue size while a transition probability is either the departure rate or the arrival rate of a data frame. The second model builds a relationship between a specific MAC protocol's behavior and the departure rate of a data frame over a given channel. For the X-MAC/BEB model, this paper extends these models of [10] to include the behavior of the BEB algorithm in addition to X-MAC protocol.

III. X-MAC/BEB PROTOCOL

X-MAC/BEB combines a BEB mechanism with X-MAC to alleviate the degree of collision efficiently and dynamically, especially, when WSNs are heavily congested. X-MAC/BEB adaptively disperses the transfer time of short preambles in proportion to the measured congestion.

Fig. 1 shows the flow chart of the X-MAC/BEB algorithm run by a sender where the four new modules added to X-MAC are highlighted with the broken-line boundary. Since the receiver of X-MAC/BEB executes the same code as X-MAC, the algorithm of the X-MAC/BEB receivers is not presented. When a node wakes up and has some data to deliver, it performs *preamble backoff* in which it randomly draws an integer number i from a range of $(0, W_0 - 1)$, where W_0 is the minimum CW. After that, it delays the transmission by i time slots.

At the expiration of this backoff timeout, it invokes two clear channel assessment (CCA) operations to make sure that the channel is free. These CCA steps are required since X-MAC does not freeze its timer when the channel is busy differently from IEEE 802.11. Without the two CCA steps, X-MAC would frequently end up delivering data over the occupied channel. It

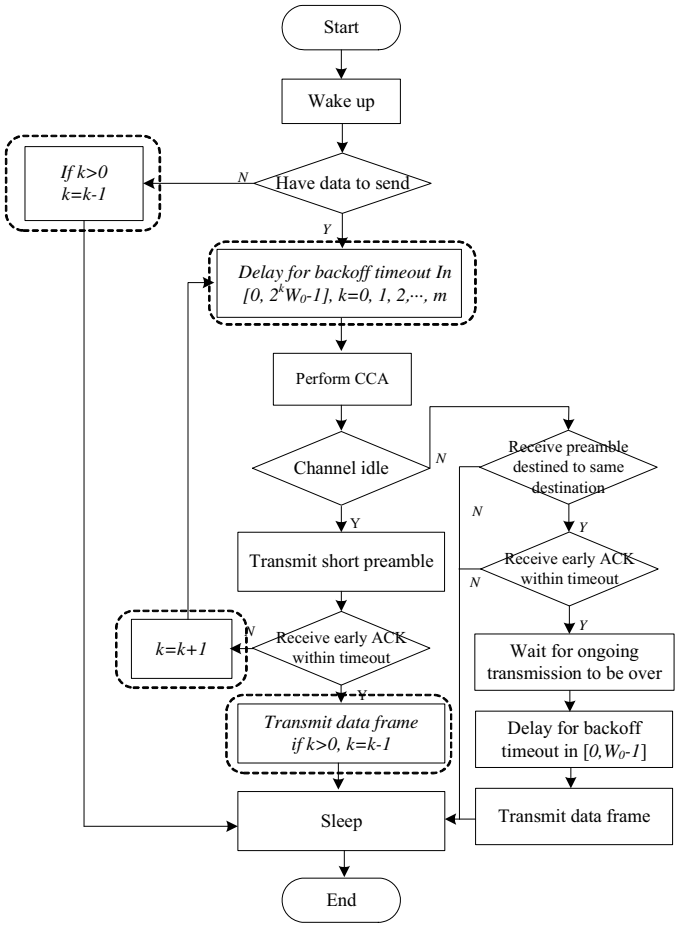


Fig. 1. Flow chart of X-MAC/BEB at sender side.

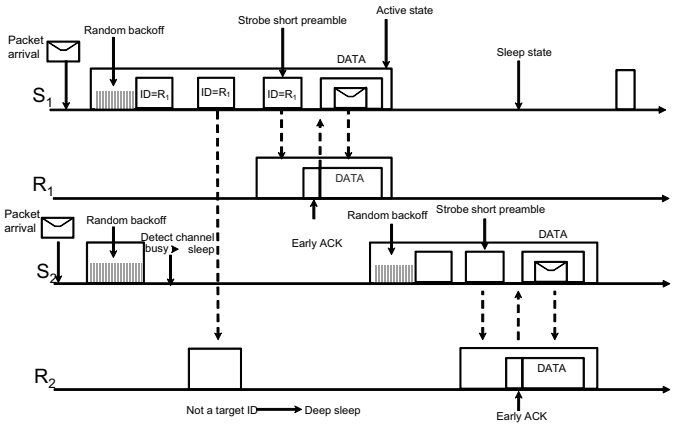


Fig. 2. Exemplary behavior of X-MAC/BEB in time domain.

also needs two CCA operations, not one, to protect acknowledgments from being disrupted since they follow their data frame after one-time slot gap.

If the channel is idle, then the node begins to send consecutively short preambles while checking for the arrival of an early ACK after each short preamble. Once the early ACK returns, it transmits one data frame and then goes back to sleep. If the early ACK does not come back within the retransmission timeout, equivalent to one period due to the collision, it increments the backoff counter k by one like the step ($k = k+1$) in Fig. 1. Then X-MAC/BEB repeats retransmission at the next cycle until either k reaches a predetermined maximum or the transmission succeeds. When the transmission is successful, it decrements k like the step ($k = k - 1$) unless $k = 0$ in Fig. 1. This is different from the traditional BEB algorithm where a successful transmission resets k to 0.

The reason for this decrement per successful transmission is that it is better to slowly change the size of CW rather than abruptly reinitialize the size of CW in WSNs where congestion may last for an extended time. Congestion in WSNs delivering surveillance high-volume multimedia data is likely to last for many duty cycles. According to [12], for example, congestion continued in a 20-node WSN for a few seconds, during which 40% of frames generated at a speed of 10 frames per second, tended to be dropped due to a busy channel. Note that the legacy BEB resets its CW under the assumption that the current network state is uncorrelated with the previous one regardless of how long ago the previous frame was sent. For this problem, variant BEB algorithms such as exponential increase exponential decrease (EIED) [18] and linear increase linear decrease (LILD) [19] algorithms do not reset CW for new transmissions like X-MAC/BEB.

If the channel is busy during the CCA operations then the sender reads the ongoing short preamble to find out the destination address, as in the step in the top condition box at the right side of Fig. 1. If the data frame is headed to the same destination, it will stay awake until the current transmission is over. After that, it calls a one-stage backoff algorithm to avoid a collision. This step is needed since multiple active nodes would probably wait during the transmission interval, resulting in frequent collisions.

If a node has no data to send at the wakeup time, it just decrements k, if k is greater than zero, rather than resets k as shown in the leftmost box in Fig. 1. The intuition for this decrement is that as time goes by, it is better to forget the previous attempts and restart transmission from the initial CW.

An exemplary behavior of X-MAC/BEB is illustrated in Fig. 2 where two senders S_1 and S_2 experiencing a collision at the last cycle, compete for winning the channel to retransmit the data frame to their receiver R_1 and R_2, respectively. Since S_1 is assumed to choose the smaller backoff timeout than S_2 after running the modified BEB algorithm, S_1 sends three short preambles once its backoff timer reaches 0. Since S_2 senses the busy channel at its CCA operation after its timer ticks down to 0, it defers its delivery to the next cycle. After waking up at the next period, S_2 restarts its BEB algorithm with the previous k since there was neither success nor failure during the previous period. Since S_2 is the only sender in this time as shown in Fig. 2, S_2 sends its two short preambles after the new backoff timeout runs down to 0.

IV. ANALYSIS OF DUTY-CYCLE MAC PROTOCOLS

To analyze precisely the performance of duty-cycle MAC protocols, Yang *et al*. [10] proposed a general framework consisting

Table 1. Important parameters.

Symbol	Description
N	Total number of nodes in the network
N_{ac}	Number of nodes woken up during T_a
n_i	Number of nodes woken up at the i^{th} time slot in T_a
a_i	Number of nodes having a data frame to send
T	Length of one cycle
T_a	Time span during which awake nodes can interfere the transmission at t
T_{un}	Time span during which awake nodes waked cannot affect the transmission at t
Q	Size of queue in MAC protocols
S	Size of a data frame
λ	Average data frame arrival rate at MAC layer
W_0	Initial CW size in unit of time slots
W_m	Maximum CW size, namely $W_m = 2^m W_0$
m	Maximum allowable back-offs
τ	Length of one time-slot
π_0	Stationary probability of the empty queue state
p	Probability of transmissions $p = P_{Succ} + P_{Coll}$
π_i	Stationary probability of state i
A_k	Probability that k frames are generated in T, $A_k = e^{-\lambda T}(\lambda T)^k/k!$ in case of Poisson
$A_{\geq k}$	Probability of no less than k frames are generated in T $A_{\geq k} = 1 - \sum_{i=0}^{k-1} A_i$

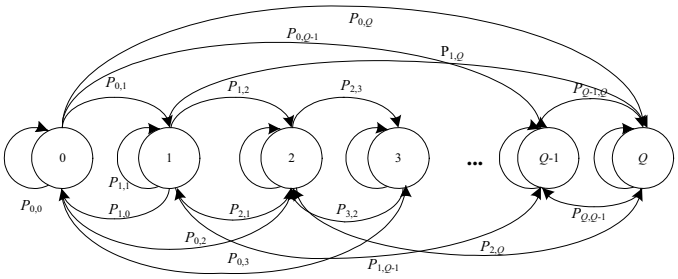


Fig. 3. Markov model of duty cycle MAC protocols' queueing behavior.

of two separate models, one for the duty cycle behavior and another for the behavior specific to a particular MAC protocol such as X-MAC and S-MAC. The first model predicts the number of frames in the buffer that arrives from the upper layer or departs to the underlying link. The second one abstracts the behavior of a particular MAC protocol to carry frames stored in the buffer down to its destination. Without the first model, the second model could not accurately evaluate the duty cycle MAC protocols operating differently from ordinary MAC protocols such as IEEE 802.11. Fig. 3 shows the first Markov model of duty cycle MAC protocols [10] describing transitions among states of a node. In Fig. 3, a node's state is denoted as a number in a circle that represents the number of frames buffered in the queue in MAC protocols. Events depicted as arrows correspond to either a frame's departure onto the link or a frame's arrival from the upper layer, respectively. As allowable transitions among states, this model assumes that only one frame is sent during one cycle whereas multiple frames can be generated within one cycle. For the sake of convenience, Table 1 lists important parameters and their meaning that will appear in the following equations.

This duty-cycle model relates three probabilities, A_i the probability for generating i-frame during one cycle, π_0 the stationary probability of an empty queue, and p the probability of a transmission at a given time. Equation (1) shows a set of equations, expressing $P_{i,j}$ a transition probability between two states in Fig. 3 with A_i and p where the subscript i and j mean the number of queued frames in the previous cycle and the number of frames in the current cycle, respectively. The first equation in (1), indicates that the transition from an empty queue to i queued frames takes place when i frames are generated within one cycle. For more details, refer to [10] .

$$\begin{cases} P_{0,i} & = A_i, i = 0, 1, \cdots, Q-1 \\ P_{0,Q} & = A_{\geq Q} \\ P_{i,i-1} & = p.A_0, i = 1, \cdots, Q-1 \\ P_{i,j} & = p \cdot A_{j-i+1} + (1-p) \cdot A_{j-1}, \\ & \quad i = 1, \cdots, Q-1, j = i1, \cdots, Q-1 \\ P_{i,Q} & = p \cdot A_{\geq Q-i+1} + (1-p) \cdot A_{\geq Q-1}, \\ & \quad i = 1, \cdots, Q-1 \\ P_{i,j} & = 0, \\ & \quad i = 2, \cdots, Q-1, j = 0, \cdots, i-2 \end{cases} \tag{1}$$

The analytical model in [10] builds (2) and (3) associating $\pi = (\pi_0, \pi_1, \pi_2, \cdots, \pi_Q)$ the state vector and P the transition probability matrix. Here π_i is the state probability that i frames are queued while the element at the i^{th} row and j^{th} column of P is $P_{i,j}$. Equation (2) says that a node should stay at one of Q possible states, where Q is the queue's maximum size.

$$\sum_{s_i \epsilon S} \pi_i = 1 \tag{2}$$

Equation (3) represents a condition for the node's state to be stable. By combining (1), (2), and (3) to remove $P_{i,j}$, π_i is solely expressed with A_i and p.

$$\pi P = \pi \tag{3}$$

If the frame arrival process is given as a closed-form equation for A_i as with the Poisson process, (4) is further built to express π_0 with a function of p [10].

$$\pi_0 = f(p) \tag{4}$$

To solve (4), we need another (5) that relates p to π_0. For (5), the second model for an MAC protocol is needed since p depends on the behavior of the specific MAC protocol. Section V presents a specific instance of (5) involving p and π_0 according to the behavior of X-MAC/BEB algorithm.

$$p = f(\pi_0) \tag{5}$$

V. PERFORMANCE ANALYSIS MODEL FOR X-MAC/BEB

This section presents analytical equations for throughput, delay, and energy efficiency of X-MAC/BEB protocol. Fig. 4 shows the behavior of both X-MAC and X-MAC/BEB during one cycle T, the top one for X-MAC [10] and the bottom one for X-MAC/BEB when a node, out of N nodes, tries to send a frame at a particular transmission time t.

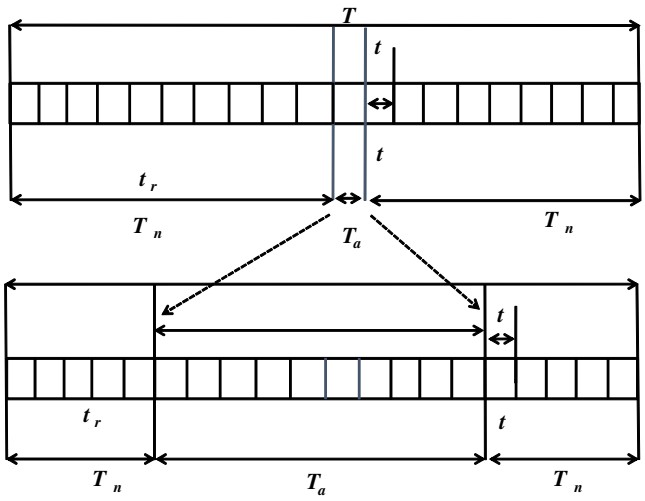


Fig. 4. Time-diagram of X-MAC and X-MAC/BEB behavior during T.

Fig. 4 emphasizes that the distinction between the two models lies in the time span T_a affecting the transmission triggered at time t. X-MAC/BEB extends T_a from a one-time slot τ in case of X-MAC to $W_k\tau = 2^k W_0\tau$ time slots to account for the effect of the BEB algorithm. Namely, T_a of X-MAC/BEB grows to $2^k W_0\tau$ where τ, k, and W_0 are the sizes of one-time slot equivalent to one tick of the BEB backoff timer, the number of CW's back-off stages or collisions, and the initial CW size, respectively. This prolonged affecting interval is because that X-MAC/BEB can put off the transmission of a frame by up to $W_k\tau$ time slots, where k ranges from 0 to m and m is the maximum number of allowable back-offs before quitting retransmission. In other words, the transmission at t can be disrupted in X-MAC/BEB by nodes awoken $W_k\tau$ ahead of t.

Fig. 4 assumes that T_a contains N_{ac} active nodes out of the total nodes N, divided into a sequence of n_i nodes activated at each i^{th} time slot in T_a while the remaining nodes $(N - N_{ac} - 1)$ wake up during T_{un}. The time t is the end of the 0^{th} time slot of T_a, at which a node is sure to send its frame. For successful transmission, X-MAC/BEB requires N_{ac} nodes awaken during T_a, not to interfere with the transmission at t.

A. Throughput Analysis of X-MAC/BEB

The average throughput of X-MAC/BEB is computed by (6) in [10], which divides the total bits successfully sent during one cycle T by T. In (6), S, π_0, and P_{Succ} represent the frame size, the probability that a given active node has no frame in the queue, and the probability of a node's successful transmission at t, respectively.

For the purpose of clear comparison, we maintained our notations in all the following equations similar to [10]. To clarify the relationship between functions, a called function keeps the same name as its caller function except that the called function adds one additional suffix.

$$THR = \frac{N \cdot (1 - \pi_0) \cdot P_{Succ} \cdot S}{T} \tag{6}$$

Table 2. Important probabilities.

Symbol	Description
P_{Succ}	Probability that a given transmission at t is successful
P_{Coll}	Probability that a given transmission at t is collides
$P_s(W_k)$	P_{Succ} under a specific case where $T_a = W_k$
$P_{s/T_a}(N_{ac})$	$P_s(W_k)$ under a specific case that N_{ac} nodes wake up during T_a
$P_{dis}(n_0, \cdots, n_{W_k-1}, N_{ac})$	Probability that $n_0, \cdots, n_{W_k-1}$ nodes are dispersed over their corresponding time slots in T_a
$P_{s/T_a/dis/i(a_i)}$	Probability that a_i nodes awake at i^{th} time slot have data to send and do not interfere the transmission
$Pdis(v_{i\_j}, \cdots, v_{i\_m}, a_i)$	Probability that $v_{i\_j}, \cdots, v_{i\_m}$ out of a_i nodes are distributed over m backoff stages
$P_{s/T_a/i}(v_{i\_j}, \cdots, v_{i\_m})$	Probability that $v_{i\_j}, \cdots, v_{i\_m}$ nodes staying at $j, \cdots, m$ backoff stages do not collide the transmission at t
$P_{FrCh}(W_k)$	Probability that the channel is free at t when T_a is W_k
$E_{free}(W_k)$	Average time span during which the channel is free when T_a is W_k
$E_{busy}(W_k)$	Average time span during which the channel is busy when T_a is W_k
$P_{free}(nT, t_{fr}, W_k)$	Probability that the channel is free for n cycles and t time slots
$P_{busy}(t_{fr}, n_{fr}, W_k)$	Probability that the channel is busy after n cycles and t time slots
$P_{busy}/T_a(N_{ac})$	Probability that the channel is busy at t by either successful transmission or collision when N_{ac} nodes are awake during T_a
$P_{busy}^{succ}(nT, t_{fr}, W_k)$	Probability that only one node sends at t after the channel is free during n cycles and t slots when T_a is W_k
$P_{busy}^{succ}(t_{fr}, n_{fr}, W_k)$	Probability that only one node sends at t after the channel is free during t slots when T_a is W_k
$P_{busy}^{coll}(nT, t_{fr}, W_k)$	Probability that collision occurs at t after the channel is free during n cycles and t slots when T_a is W_k
$P_{busy}^{coll}(t_{fr}, n_{fr}, W_k)$	Probability that collision occurs at t

A.1 *Probability of a Successful Transmission at t*

Equation (7) computes P_{Succ} as an average of the product of two probabilities, $P_s(W_k)$ the probability that only one node transmits its frame at t and $P_{FrCh}(W_k)$ the probability that the channel is free at t. For averaging the product of two probabilities over all possible T_a, (7) multiplies $1/(1 + m)$ with $P_s(W_k) \cdot P_{FrCh}(W_k)$ and then sums the product over k from 0 to m. Note that $1/(1 + m)$ is the probability that T_a is W_k since the subscript k can be any number from 0 to m.

$$P_{Succ} = \frac{1}{(1 + m)} \sum_{k=0}^{m} P_s(W_k) \cdot P_{FrCh}(W_k) \tag{7}$$

As a summary of all the probabilities to be calculate in the below, Table 2 pairs the important probabilities and their meaning.

Equation (8) equals $P_s(W_k)$ to an average of the successful transmission probability over two variables: the transmission time t that can be any slot from W_k to T and N_{ac} that ranges

from 0 to $(N-1)$. Since the starting point of the time window T_a can slide from 0 to W_k in Fig. 4, t can be between 1 to $(W_k - T)$ in Fig. 4. The inner term of the two summations in (8), the successful transmission probability of a specific case where W_k and N_{ac} are given, is decomposed into four factors. These are first, the number of ways to pick N_{ac} out of $(N-1)$, second, the probability that N_{ac} nodes awake during T_a, third, $P_s/T_a(N_{ac})$ the probability that the active N_{ac} nodes, never interfere with the transmission at t, and finally, fourth, the probability that the remaining $N-1-N_{ac}$ nodes wake up during T_{un}, respectively. Note that since a node randomly wakes up anytime during T, the probability that it wakes during an interval i out of T is simply i/T.

Compared to its corresponding equation for X-MAC [10], (8) includes three noticeably different factors: $T - W_k$ the time span to average which was T in X-MAC [10], $((T-W_k)/T)^{N-1-N_{ac}}$ the probability that $(N-1-N_{ac})$ nodes are active in T_{un}, which was $((T-1)/T)^{N-1-i}$ in X-MAC, and finally $P_s/T_a(N_{ac})$ the probability that N_{ac} active nodes in T_a never disrupt the transmission at t, which was $(1/T)^i \cdot (\pi_0)^i$ in X-MAC. In X-MAC, $(1/T)^i \cdot (\pi_0)^i$ denotes the probability that i nodes that are awake at a given time slot have no data, which is the sufficient condition for no interference at t.

$$P_s(W_k) = \sum_{t=1}^{T-W_k} \frac{1}{T-W_k} \sum_{N_{ac}=0}^{N-1} \binom{N-1}{N_{ac}} \cdot \left(\frac{W_k}{T}\right)^{N_{ac}} \cdot P_{s/T_a}(N_{ac}) \cdot \left(\frac{T-W_k}{T}\right)^{N-1-N_{ac}} \tag{8}$$

In (9) $P_{s/T_a}(N_{ac})$ averages a product of two probabilities $P_{dis}(n_0, ., n_{W_k-1}, N_{ac})$ and $P_{s/T_a/dis}(n_0, .n_{W_k-1}, W_k)$ over n_i, the number of nodes awakened in the i^{th} slot of T_a varied from 0 to N_{ac}. $P_{dis}(n_0, \cdots, n_{W_k-1}, N_{ac})$ is the distribution probability that $n_0, \cdots, n_{W_k-1}$ nodes are spread over W_k time slots, namely the 0^{th}, the 1^{st},..,the $W_k^{\text{th}} - 1$ slot in T_a as shown in Fig. 4. $P_{s//Ta/dis}(n_0, \cdots, n_{W_k-1}, W_k)$ is the probability that $n_0, \cdots, n_{W_k-1}$ nodes distributed in T_a do not hamper the transmission at t.

$$P_{s/T_a}(N_{ac}) = \sum_{n_{W_k-1}=0}^{N_{ac}-n_0\cdots n_{W_k-2}} \cdots \sum_{n_0=0}^{N_{ac}} P_{dis}(n_0, \cdots, , n_{W_k-1}, N_{ac}) \cdot P_{s/T_a/dis}(n_0, \cdots, n_{W_k-1}, W_k) \tag{9}$$

Equation (10) seeks $P_{dis}(n_0, \cdots, n_{W_k-1}, N_{ac})$ as a ratio of the number of ways to arrange $n_0, \cdots, n_{W_k-1}$ nodes over W_k slots of T_a in Fig. 4 to the total number of ways to disperse N_{ac} nodes over W_k slots. For (10), we assume that nodes are uniformly distributed over T_a. Fig. 5 plots the average number of nodes active at each slot in T_a collected from 15 ns-2 simulations of a 30-node WSN. Fig. 5 illustrates that one node on average tends to wake up at each slot with $m = 2$, which implies that T_a is $32 = 2^2 \cdot W_0$ time slots since W_0 is 8.

$$P_{dis}(n_0, \cdots, n_{W_k-1}, N_{ac}) = \frac{n_0!\cdot, \cdots, \cdot n_{W_k-1}!}{N_{ac}!} \tag{10}$$

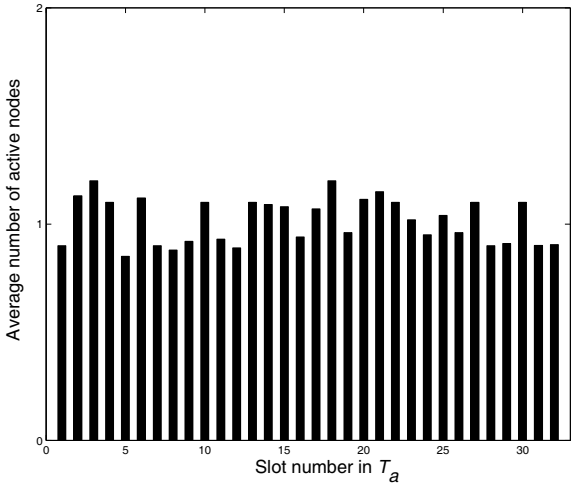


Fig. 5. Node distribution at each slot in T_a.

$P_{s/T_a/dis}(n_0, \cdots, n_{W_k-1}, W_k)$ in (11) is factorized into W_k factors, the i^{th} factor of which is the average probability that n_i nodes activated at the i^{th} time slot do not obstruct the transmission at t. The no-interference probability by the nodes at i^{th} slot is further divided into four factors. These include first, the number of ways to pick a_i out of n_i, second, the probability that $(n_i - a_i)$ nodes have no data to send, third, the probability that a_i nodes have data to send, forth, $P_{s/T_a/dis/i}(a_i)$ the probability that the backoff timer of all a_i nodes should be expired after t.

$$P_{s/T_a/dis}(n_0, \cdots, n_{W_k-1}, W_k) = \prod_{i=0}^{W_k-1} \sum_{a_i=0}^{n_i} \binom{n_i}{a_i} \cdot \pi_0^{n_i-a_i} \cdot (1-\pi_0)^{a_i} \cdot (P_{s/T_a/dis/i}(a_i))^{a_i} \tag{11}$$

$P_{s/T_a/dis/i}(a_i)$ in (12) averages the product of two probabilities $P_{dis}(v_{i\_j}, \cdots, v_{i\_m}, a_i)$ and $P_{s/T_a/i}(v_{i\_j}, \cdots, v_{i\_m})$ over $v_{i\_j}, \cdots, v_{i\_m}$ each of which varies from 0 to a_i. $P_{dis}(v_{i\_j}, \cdots, v_{i\_m}, a_i)$ is the probability that $v_{i\_j}, \cdots, v_{i\_m}$ nodes stay at the j^{th}, $(j+1)^{\text{th}}, \cdots$, and the m^{th} backoff stages while $P_{s/T_a/i}(v_{i\_j}, \cdots, v_{i\_m})$ is the possibility that these nodes have their backoff timer expired after t, not to contaminate the transmission at t. Namely, the argument v_{i_l} of these two probabilities represents the number of nodes which suffered l collisions. Here l ranges from j to m and j is the least integer satisfying $2^j > i$. The reason why l starts from j not 0 is that the CW size of any node among v_{i_l} should be larger than i, the time distance between the i^{th} time slot and t so that the backoff timers of v_{i_l} nodes are expired later than t.

$$P_{s/T_a/dis/i}(a_i) = \sum_{v_{i\_m}=0}^{a_i-v_{i\_j}\cdots v_{i\_m-1}} \cdots \sum_{v_{i\_j}=0}^{a_i} P_{dis}(v_{i\_j}, \cdots, v_{i\_m}, a_i) \cdot P_{s/T_a/i}(v_{i\_j}, \cdots, v_{i\_m}) \tag{12}$$

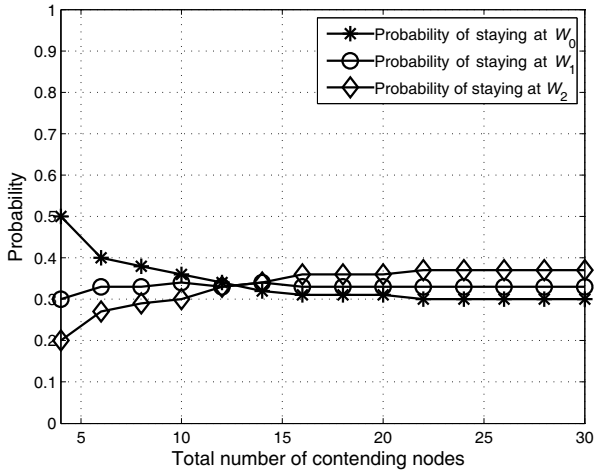


Fig. 6. Probability that a node in T_a stays at a given backoff stage.

Equation (13) calculates $P_{dis}(v_{i\_j}, \cdots, v_{i\_m}, a_i)$ as (10) under the assumption that a_i nodes are uniformly distributed over $(m-j)$ backoff stages.

$$P_{dis}(v_{i\_j}, \cdots, v_{i\_m}, a_i) = \frac{v_{i\_j}! \cdot, \cdots, \cdot v_{i\_m}!}{a_i!} \tag{13}$$

As a verification of this uniform distribution assumption for (13), Fig. 6 depicts the probability that a node stays at any of three backoff stages, W_0, W_1, and W_2 as a function of the total number of contending nodes. To plot each graph in Fig. 6, we count the number of collisions suffered by nodes awaken at each time slot in T_a. Each point in Fig. 6 stands for the average of five ns-2 simulations with $m = 2, W_0 = 8$, and $T = 100$ ms. Fig. 6 indicates that the three probabilities of nodes residing at W_0, W_1, and W_2 approximately converge to 1/3 as the number of contending nodes increases. When contending nodes are fewer than 10, however, the probability of staying at W_0 is higher than the other two probabilities due to the negligible collision rate. Based on this observation, (13) would only be valid for overcrowded WSNs for which we propose the X-MAC/BEB algorithm.

$P_{s/T_a/i}(v_{i\_j}, \cdots, v_{i\_m})$ in (14) consists of $(m-j)$ factors, each of which is the probability that v_{i_l} nodes at the l^{th} backoff stage do not clash with the transmission at t. Namely, each factor represents the probability that the backoff timeout of all v_{i_l} is larger than i, the time distance from t to the i^{th} time slot. Furthermore, the probability that a node out of v_{i_l} has its backoff timeout larger than i is $(W_l - i)/W_l$.

$$P_{s/T_a/i}(v_{i\_j}, \cdots, v_{i\_m}) = \prod_{l=j}^{m} \left(\frac{W_l - i}{W_l}\right)^{v_{i\_l}} (i \le W_j) \tag{14}$$

A.2 *Probability of Collision at t*

As shown in (15) P_{Coll} the probability of the transmission colliding at t averages the product of $P_f(W_k)$ and $P_{FrCh}(W_k)$, which is the probability that one or more nodes in T_a sends their frames at t as like P_{Succ} in (7).

$$P_{Coll} = \frac{1}{(1+m)} \sum_{k=0}^{m} P_f(W_k) \cdot P_{FrCh}(W_k) \tag{15}$$

Differently from $P_s(W_k)$ in (8),$P_f(W_k)$ in (16) includes one additional probability $\binom{N_{ac}}{1}(1-\pi_0) \cdot (1/W_{avg}(W_k))$ that at least one node out of N_{ac} nodes in T_a sends its frame at t to incur collision at t. Here, these three factors refer to the number of ways to pick one sender out of N_{ac}, (1-π_0), the probability of non-empty queue at that sender, and 1/$W_{avg}(W_k)$ the probability that the backoff timer of that sender expires at t to collide the transmission at t, respectively. Note that the probability of timeout at t is simply the inverse of the average maximum CW size.

$$P_f(W_k) = \sum_{t=1}^{T-W_k} \frac{1}{T-W_k} \sum_{N_{ac}=1}^{N-1} \binom{N-1}{N_{ac}} \cdot \left(\frac{W_k}{T}\right)^{N_{ac}} \cdot \binom{N_{ac}}{1}(1-\pi_0) \cdot \left(\frac{1}{W_{avg}(W_k)}\right) \cdot P_{s/T_a}(N_{ac}-1) \cdot \left(\frac{T-W_k}{T}\right)^{N-1-N_{ac}} \tag{16}$$

Equation (17) computes $W_{avg}(W_k)$, the average size of W_k where W_k indicates the maximum CW that a node in T_a can keep. For $W_{avg}(W_k)$, at first, remember that the nodes in T_a should have W_k at least larger than the time distance between t and their awake time slot. That is the necessary condition to have a collision occur exactly at t. Otherwise, there is no way for a transmission to start after t. A node in the time slots from W_{x-1} to $(W_x - 1)$ in T_a in Fig. 4, for example, should have CW whose maximum size is between W_x and W_x including them. For this computation, we divide T_a into intervals such as $(W_{-1}, W_0 - 1), (W_0, W_1 - 1), (W_{x-1}, W_x - 1), \cdots, (W_{k-1}, W_k - 1)$. Therefore, the average maximum CW size of a node in time slots from W_{x-1} to $(W_x - 1)$ is $(W_x + W_{x-1} + \cdots + W_k)/(k - x + 1)$. Since the probability that a node wakes up in this interval $(W_{x-1}, W_x - 1)$ is $(W_x, W_{x-1})/W_k$, (17) multiplies the average size of the maximum CW size in this interval with its weight $(W_x, W_{x-1})/W_k$ and finally adds this product over all the intervals in T_a.

$$W_{avg}(W_k) = \left(\sum_{x=0}^{k} \left(\frac{1}{k-x+1} \sum_{y=x}^{k} W_k\right) \cdot \frac{(W_x - W_{x-1})}{W_k}\right), \quad \text{where } W_{-1} = 0 \tag{17}$$

$P_{f/T_a}(N_{ac}-1)$ in (16) is drawn in the same way as $P_{s/T_a}(N_{ac})$ in (8) except that $P_{s/T_a/i}(v_{i\_j}, \cdots, v_{i\_m})$ in (18) includes one more case where the backoff timer precisely expires at t. Note those other nodes among $(N_{ac}-1)$ nodes are also allowed to send their frames at t since a collision will occur at t. For this, $(W_l - i)$ the numerator of $P_{s/T_a/i}(v_{i\_j}, \cdots, v_{i\_m})$ in

(14) is incremented like $(W_l - i + 1)$ in (18).

$$P_{f/T_a/i}(v_{i\_j}, \cdots, v_{i\_m}) = \prod_{l=j}^{m} \left(\frac{W_l - i + 1}{W_l} \right)^{v_{i\_l}} (i \le W_j) \tag{18}$$

A.3 *Probability of the Channel Being Free*

Equation (19) derives $P_{FrCh}(W_k)$ as the ratio of $E_{free}(W_k)$ the duration of the free channel to the entire channel time, simply the sum of $E_{free}(W_k)$ and $E_{busy}(W_k)$. $E_{free}(W_k)$ amounts to the average free time span between two adjacent busy intervals while $E_{busy}(W_k)$ denotes the average time consumed by either collision or a successful transmission.

$$P_{FrCh}(W_k) = \frac{E_{free}(W_k)}{E_{free}(W_k) + E_{busy}(W_k)} \tag{19}$$

Equation (20) seeks $E_{free}(W_k)$ by taking an average of a free channel time $(nT + t_{fr})$ multiplied by its probability $P_{free}(nT, t_{fr}, W_k)$ over n and t_{fr}. These two variables represent the number of free channel cycles ahead of the current cycle and the number of the free time slot in the current cycle, respectively.

$$E_{free}(W_k) = \sum_{n=0}^{\infty} \sum_{t_{fr}=0}^{T-1} (nT + t_{fr}) \cdot P_{free}(nT, t_{fr}, W_k) \tag{20}$$

For $P_{free}(nT, t_{fr}, W_k)$, the first factor π_0^{Nn} in (21) denotes the probability that all N nodes have no data during consecutive nT cycles while the remaining factors are the probability that the channel is idle until $t = t_{fr} + W_k$ as shown in Fig. 4. The remaining factors are grouped into two probabilities, first that n_{fr} nodes wake up during t_{fr} but with an empty queue and second $P_{busy}(t_{fr}, n_{fr}, W_k)$ that at least one node among $(N - n_{fr})$ nodes awaken during T_a starts the transmission at t. Thus, $P_{busy}(t_{fr}, n_{fr}, W_k)$ is the probability that a successful transmission or collision occurs at t after the idle period of $(t_{fr} + W_k)$.

$$P_{free}(nT, t_{fr}, W_k) = \pi_0^{Nn} \sum_{n_{fr}=0}^{N-1} \binom{N}{n_{fr}} \left(\frac{t_{fr}}{T} \right)^{n_{fr}} \cdot \pi_0^{t_{fr}} \cdot P_{busy}(t_{fr}, n_{fr}, W_k) \tag{21}$$

$P_{busy}(t_{fr}, n_{fr}, W_k)$ in (22) is built in a similar way as $P_f(W_k)$ in (16) except for the different time span over which they are computed. The time span for $P_{busy}(t_{fr}, n_{fr}, W_k)$ is $(T - t_{fr})$ the remaining interval except the first unaffected time in Fig. 4 while that for $P_f(W_k)$ is the entire cycle T. This distinction leads to three differences between (16) and (22). First is that $P_f(W_k)$ takes one more average over T_a. Second is that the total number of nodes for $P_{busy}(t_{fr}, n_{fr}, W_k)$ is $(N - n_{fr})$ not $(N - 1)$. Third is that the unaffected interval for $(N - n_{fr})$ is $(T - W_k - t_{fr})$ not $(T - W_k)$. $P_{busy/T_a}(N_{ac} - 1)$ in (22) is calculated in the same way as $P_{s/T_a}(N_{ac})$ in (9) except $P_{s/T_a/i}(v_{i\_j}, \cdots, v_{i\_m})$ in (14) to account for collision cases.

$$P_{busy}(t_{fr}, n_{fr}, W_k) = \sum_{N_{ac}=1}^{N-n_{fr}} \binom{N - n_{fr}}{N_{ac}} \cdot \left(\frac{W_k}{T} \right)^{N_{ac}} \cdot \binom{N_{ac}}{1} (1 - \pi_0) \cdot \left(\frac{1}{W_{avg}(W_k)} \right) \cdot P_{busy/T_a}(N_{ac} - 1) \cdot \left(\frac{T - W_k - t_{fr}}{T} \right)^{N - N_{ac} - n_{fr}} \tag{22}$$

For $P_{busy/T_a/i}(N_{ac} - 1)$, $P_{s/T_a/i}(v_{i\_j}, \cdots, v_{i\_m})$ in (14) should be replaced with $P_{busy/T_a/i}(v_{i\_j}, \cdots, v_{i\_m})$ in (23). The numerator of the inner term in (23) is incremented from $(W_l - i)$ to $(W_l - i + 1)$ to allow one more collision case, where the backoff timer goes off exactly at t.

$$P_{busy/T_a/i}(v_{i\_j}, \cdots, v_{i\_m}) = \prod_{l=j}^{m} \left(\frac{W_l - i + 1}{W_l} \right)^{v_{i\_l}} (i \le W_j) \tag{23}$$

Equation (24) calculates $E_{busy}(W_k)$ as the average time of the busy channel by adding two terms, the average successful transmission time and the average collision time. The successful transmission time is further divided into two, the half period $T/2$ wasted on average for the receiver to switch on and the transmission delay t_{Data}. The second inner term T represents the time wasted for collision since the sender cannot recognize collision so that it keeps sending short preambles during the whole cycle.

$$E_{busy}(W_k) = \sum_{n=0}^{\infty} \sum_{t_{fr}=0}^{T-1} \left(\frac{T}{2} + t_{Data} \right) \cdot P_{busy}^{succ}(nT, t_{fr}, W_k) + T \cdot P_{busy}^{coll}(nT, t_{fr}, W_k) \tag{24}$$

Equation (25) computes $P_{busy}^{succ}(nT, t_{fr}, W_k)$ the probability of successful transmission at t after $(t_{fr} + W_k)$ as in (21) but excludes the collision case.

$$P_{busy}^{succ}(nT, t_{fr}, W_k) = \pi_0^{Nn} \sum_{n_{fr}=0}^{N-1} \binom{N}{n_{fr}} \left(\frac{t_{fr}}{T} \right)^{n_{fr}} \cdot \pi_0^{t_{fr}} \cdot P_{busy}^{succ}(t_{fr}, n_{fr}, W_k) \tag{25}$$

$P_{busy}^{succ}(t_{fr}, n_{fr}, W_k)$ is derived in the same way as in (22) except $P_{busy/T_a/i}^{succ}(v_{i\_j}, \cdots, v_{i\_m})$ in (26) corresponding to $P_{busy/T_a/i}(v_{i\_j}, \cdots, v_{i\_m})$ in (23). Namely, the number of cases belonging to is $(W_l - i)$ decremented from $(W_l - i + 1)$ in (26) to rule out the collision case.

$$P_{busy/T_a/i}^{succ}(v_{i\_j}, \cdots, v_{i\_m}) = \prod_{l=j}^{m} \left(\frac{W_l - i}{W_l} \right)^{v_{i\_l}} (i \le W_j) \tag{26}$$

Equation (27) establishes $P_{busy}^{coll}(nT, t_{fr}, W_k)$ the probability that collision occurs at t, in the same way as (25) except that it

includes only collisions

$$P_{busy}^{coll}(nT, t_{fr}, W_k) = \pi_0^{Nn} \sum_{n_{fr}=0}^{N-2} \binom{N}{n_{fr}} \left(\frac{t_{fr}}{T}\right)^{n_{fr}} \cdot \pi_0^{t_{fr}} \cdot P_{busy}^{coll}(t_{fr}, n_{fr}, W_k) \tag{27}$$

Equation (28) computes $P_{busy}^{coll}(t_{fr}, n_{fr}, W_k)$ in the same way as $P_{busy}(t_{fr}, n_{fr}, W_k)$ in (22) except that it substitutes $\binom{N_{ac}}{1}(1-\pi_0)\cdot(1/W_{avg}(W_k))$ the probability of one transmission at t with $\binom{N_{ac}}{2}(1-\pi_0)^2\cdot(1/W_{avg}(W_k))^2$the probability of two simultaneous transmissions at t to make sure of the collision at t. Other than that $P_{busy}^{coll}(t_{fr}, n_{fr}, W_k)$ differs from $P_{busy}(t_{fr}, n_{fr}, W_k)$ in that the summation over N_{ac} starts from two not one.

$$P_{busy}^{coll}(t_{fr}, n_{fr}, W_k) = \sum_{N_{ac}=2}^{N-n_{fr}} \binom{N-n_{fr}}{N_{ac}} \cdot \left(\frac{W_k}{T}\right)^{N_{ac}} \cdot \binom{N_{ac}}{2}(1-\pi_0)^2 \cdot \left(\frac{1}{W_{avg}(W_k)}\right)^2 \cdot P_{busy/T_a}(N_{ac}-2) \cdot \left(\frac{T-W_k-t_{fr}}{T}\right)^{N-N_{ac}-n_{fr}} \tag{28}$$

B. Delay Analysis of X-MAC/BEB

Equation (29) computes the whole delay taken by a frame, from its generation to its successful delivery to the destination. It is broken down into three components, D_Q queueing delay, D_{CW} random backoff delay, and D_C contention delay.

$$D = D_Q + D_{CW} + D_C \tag{29}$$

In detail, D_Q represents the average waiting time of a frame at the buffer while D_C refers to the delay spent for a frame, to be finally delivered to the receiver after undergoing contention. Since D_Q and D_C in the X-MAC/BEB algorithm are same as those of X-MAC, please refer to [10] for more details about them. In (30), D_{CW} the extra delay introduced by the BEB algorithm sums the half of the CW's size at each backoff stage from 0 to m and then averages the sum by the total number of stages.

$$D_{CW} = \frac{1}{(1+m)} \sum_{k=0}^{m} \left(\frac{W_k \cdot \tau}{2}\right) \tag{30}$$

C. Energy Consumption Analysis of X-MAC/BEB

Equation (31) calculates E the average total energy consumption during one cycle by a node, as the sum of energy at five states: E_1 at the successful transmission state, E_2 at the successful reception state, E_3 at the transmission failure state, E_4 at the aborted reception state, and finally E_5 at the overhearing state. Since the receiver of X-MAC/BEB behaves in the same way as that of X-MAC, E_2, E_4, and E_5 dissipated at the receiver remain unchanged from [10].

$$E = E_1 + E_2 + E_3 + E_4 + E_5 \tag{31}$$

In (32), E_1 is divided into two factors, $(1-\pi_0)\cdot P_{Succ}\cdot\tau$ for the fraction of one-time slot taken by a successful transmission and the following factor for the energy consumed by a successful transmission. The energy of a successful transmission is further decomposed into four types of energy for running the average backoff timeout, transmitting the average number of short preambles, an early ACK, and finally a data frame, respectively. For the average backoff timeout, the first term added for X-MAC/BEB takes the average of $W_k/2$ over k from 0 to m. Here $txp, rxp, t_{pre}, t_{ACK}$, and t_{DATA} represent transmitting power, receiving power, time to transmit one short preamble, one ACK, and one data frame, respectively.

$$E_1 = (1-\pi_0)\cdot P_{Succ}\cdot\tau\left(\frac{1}{(1+m)}\sum_{k=0}^{m}\left(\frac{W_k\cdot\tau}{2}\right)rxp + \frac{T}{2}\left(\frac{t_{pre}}{t_{pre}+t_{ACK}}\right)txp + \frac{T}{2}\left(\frac{t_{ACK}}{t_{pre}+t_{ACK}}\right)rxp + t_{DATA}\cdot txp\right) \tag{32}$$

Equation (33) for E_3 is divided into two factors as (32), the fraction of one cycle wasted due to the failed transmission and the average energy dissipated during this failed transmission. The aborted transmission interval consists of the average backoff timeout and a series of pairs of one short preamble and its ACK lasting over the whole cycle length T. After sending each preamble, the sender should listen for ACK for the whole cycle before it abandons the transmission. Just like E_1 , E_3 has one more term than [10].

$$E_3 = (1-\pi_0)\cdot P_{Coll}\cdot\tau\left(\frac{1}{(1+m)}\sum_{k=0}^{m}\left(\frac{W_k\cdot\tau}{2}\right)rxp + T\left(\frac{t_{pre}}{t_{pre}+t_{ACK}}\right)txp + T\left(\frac{t_{ACK}}{t_{pre}+t_{ACK}}\right)rxp\right) \tag{33}$$

Equation (34), finally presents energy consumed per frame per node per second where S represents the frame size as defined in [10].

$$E_{frame/node/s} = \frac{E\cdot S}{THR\cdot T} \tag{34}$$

VI. EVALUATION

This section measures the throughput, delay, and energy consumption of X-MAC/BEB and X-MAC by their analytical models and the corresponding ns-2 simulations under various environments. Table 3 lists important parameters and their assigned values employed in the numerical analysis and simulations. For clearly showing the effect of BEB over X-MAC, we set all the operational parameters of X-MAC/BEB to the same value and plot graphs in the same way as in [10].

Figs. 7, 8, and 9 depicts the throughput, namely the number of bytes successfully sent during one cycle T, produced from (6)

Table 3. Values for X-MAC and X-MAC/BEB parameters.

Parameter	Value
Bandwidth	250 kbps
Transmission range	250 m
Carrier sensing range	550 m
τ	20 μs
λ	1 frame/s
T_{Active}	15 ms
t_{ACK}	1 ms
t_{pre}	3 ms
t_{DATA}	5 ms
T	50–300 ms
Q	10
S	50 bytes
txp	59.1 mW
rxp	52.2 mW

and simulations as a function of N the total number of nodes, T cycle length, and W_0 initial CW size, respectively. In this experimental network, the underlying physical link speed, W_0, m, and T for X-MAC/BEB are set to 250 kbps, 8, 2, and 100 ms, respectively. Fig. 7, at first, shows that X-MAC/BEB, denoted by the circle-ridden line, outperforms X-MAC as indicated by the star-ridden line as the WSNs become more overcrowded. When N approaches 40, for instance, X-MAC/BEB outperforms X-MAC by 40%. This improved achievement comes from a decreased collision rate, reduced from 1.45% to 0.76% owing to the BEB algorithm. The vertical bars on each line corresponding to the range of 15 simulation outcomes indicate that the analytical predictions coincide with their simulation measurements by a 3% deviation at maximum. Note that all vertical bars in the following figures correspond to the distribution of 15 simulation results.

Fig. 8 plots the effect of T on the throughput of the X-MAC and X-MAC/BEB algorithm in the 40-node network. It shows that the throughput of X-MAC/BEB drops more steeply than that of X-MAC as T increases since nodes rarely transmit data. Thus, the throughput gap between X-MAC and X-MAC/BEB almost disappears as T grows. When T equals 300 ms, for example, the gap narrows to 8% since nodes tend to sleep longer as T is larger. In other words, since nodes have fewer opportunities to send in longer T, the BEB algorithm hardly makes a big difference in throughput.

Fig. 9 describes the dependency of X-MAC/BEB throughput on W_0 and N. The two graphs in Fig. 9, representing $N = 20$ and $N = 40$, display that the throughput of X-MAC/BEB shortly improves as W_0 expands from one to two but after this point, further improvement is not observed. The comparison of two graphs verifies that W_0 tends to affect slightly the throughput depending on the network size. At $W_0 = 32$, the throughput of $N = 40$ jumps up by 10% more than that of $N = 20$.

Figs. 10–12 plot the delay of a successful transmission as measured in (29) and simulations, as a function of N, T, and W_0, respectively, under the same experiment setups as Figs. 7, 8, and 9. Fig. 10 shows that the delay of X-MAC/BEB almost linearly grows as the delay of X-MAC when the network becomes overloaded. The delay growth of these two protocols is due to frequent collisions as the network is flooded with data frames. The comparison between the two graphs in Fig. 10,

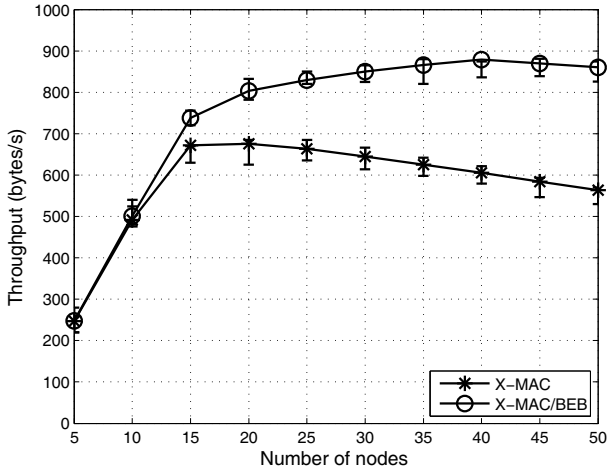


Fig. 7. Throughput vs. number of nodes.

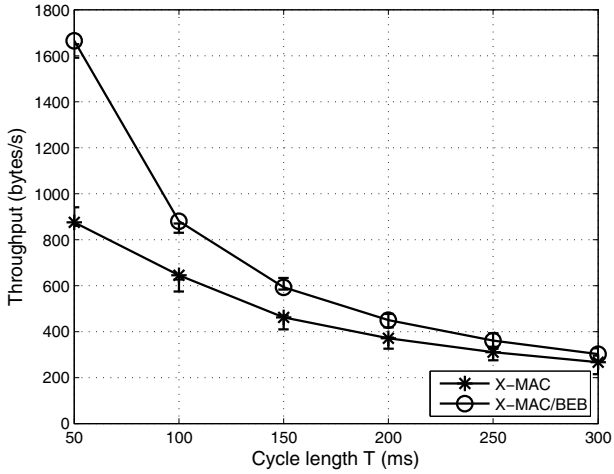


Fig. 8. Throughput vs. cycle length.

however, emphasizes that X-MAC/BEB tends to maintain less delay under congested networks than X-MAC. X-MAC/BEB, for example, shortens the delay of X-MAC by up to 15% at the 40-node network by reducing collisions.

Fig. 11 shows that X-MAC/BEB and X-MAC obviously slow down their delivery in proportion to T due to a longer sleep time. Even though their delay linearly increases at the same rate as a function of T, X-MAC/BEB still performs better than X-MAC. X-MAC/BEB, for example, cuts down the delay of X-MAC by 4% in a network with $N = 40$ and $T = 300$ ms. This small gain in delay is due to that there are not many collisions to avoid by X-MAC/BEB in this lightly loaded network.

Fig. 12 measures the delay variance as a function of W_0 under the two networks with $N = 20$ and $N = 40$ as in Fig. 9. Fig. 12 presents that these delay graphs almost flatten after a little downward slope around the small W_0 less than four. The reason why W_0 rarely affects the delay is that the X-MAC/BEB algorithm tends to remain at the average CW size under saturated networks regardless of W_0. Remember that the BEB algorithm

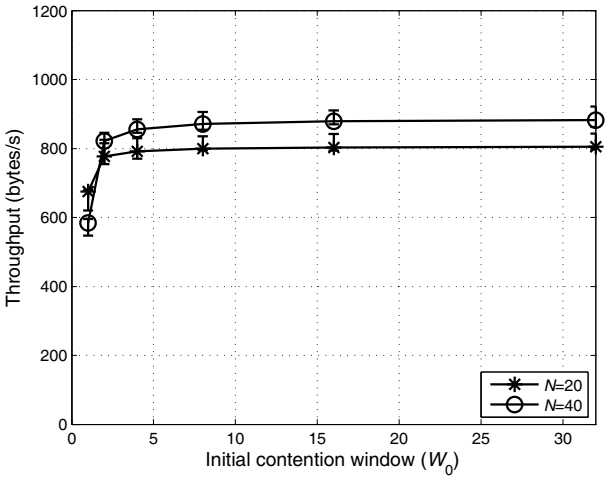


Fig. 9. Throughput vs. initial contention window.

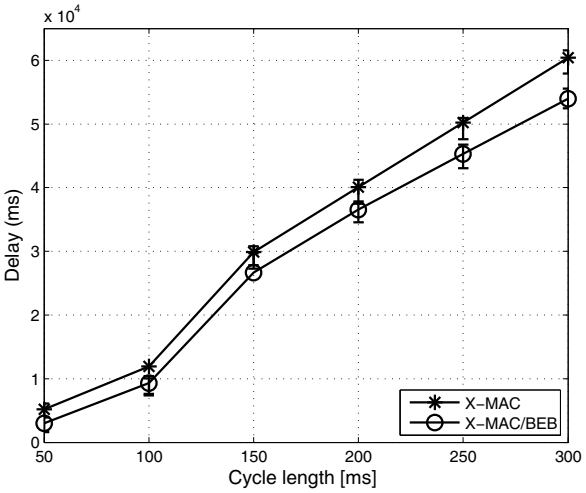


Fig. 11. Delay vs. cycle length.

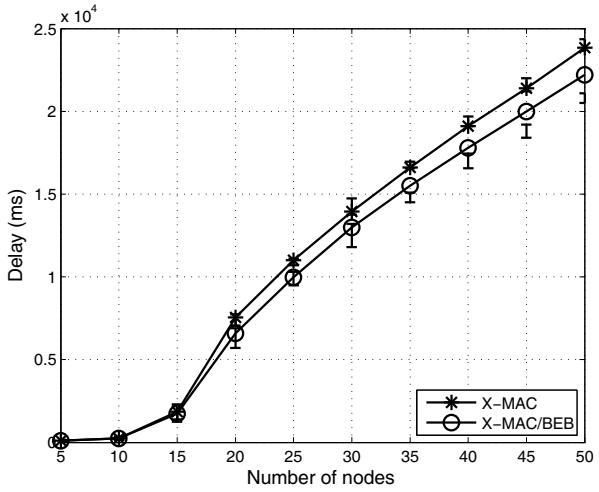


Fig. 10. Delay vs. number of nodes.

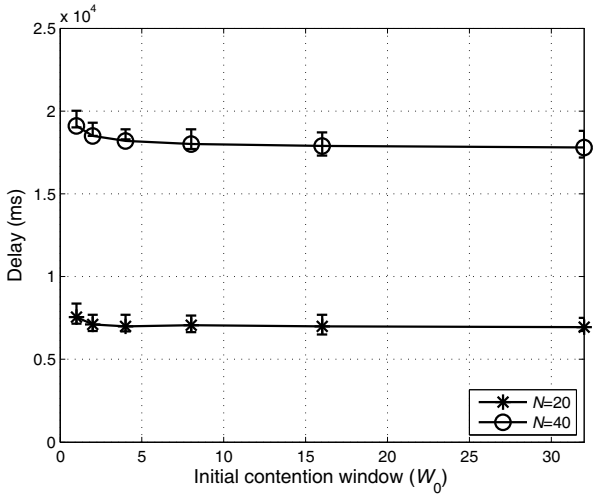


Fig. 12. Delay vs. initial contention window.

of X-MAC/BEB does not start from W_0 at each new frame's transmission. Fig. 12 also observes that the delay of $N = 40$ is almost double that of $N = 20$ since the former experiences collision 4% more often than the latter.

Figs. 13, 14, and 15 draw $E_{frame/node/s}$ as evaluated in (34) and simulations as a function of N, T, and W_0, respectively under the same experiment setups Figs. 7, 8, and 9. Fig. 13 shows that$E_{frame/node/s}$ of X-MAC/BEB and X-MAC rapidly decreases as the network is populated since nodes are less likely to send frames due to the busier channel, leading cessation of data transmission so that they go back to sleep more frequently. Note that once the channel is already taken, nodes in both X-MAC/BEB and X-MAC need to sleep until the new period starts. The gap between these two graphs, furthermore, widens as the number of nodes approaches 50 due to the reduced collision, for example, 40% less energy at a 40-node network. Fig. 13 indicates that $E_{frame/node/s}$ of X-MAC/BEB simulations deviates from that of the analytical model by 3% while their maximum standard deviation is around 4%.

Fig. 14 plots that $E_{frame/node/s}$ of X-MAC/BEB and X-MAC increases in proportion to T except for one slight dip at $T = 100$ ms in the case of X-MAC since more short preambles need to be forwarded in a longer T. $E_{frame/node/s}$ of X-MAC/BEB, for example, jumps up by 81% maximally when T extends from 50 ms to 300 ms. $E_{frame/node/s}$ of X-MAC/BEB, furthermore, is at least 48% lower than that of X-MAC due to the reduced collision rate from 1.45% to 0.76% in the 40-node network. These analytical results are validated by their corresponding simulations with 3% deviation.

Fig. 15, finally, draws $E_{frame/node/s}$ of X-MAC/BEB as a function of W_0 in the 20-node and 40-node network. Even though the change of W_0 from one to two incurs an abrupt fall in $E_{frame/node/s}$, after this point $E_{frame/node/s}$ rarely depends on W_0. The reason for this independency on W_0 is that X-MAC/BEB keeps its last CW for new transmissions rather than restarts from W_0. $E_{frame/node/s}$ of X-MAC/BEB is 40% less on average than that of X-MAC in the 40-node network.

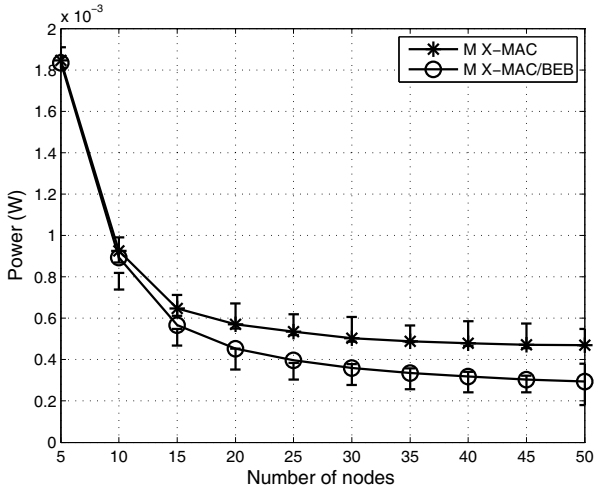


Fig. 13. Energy vs. number of nodes.

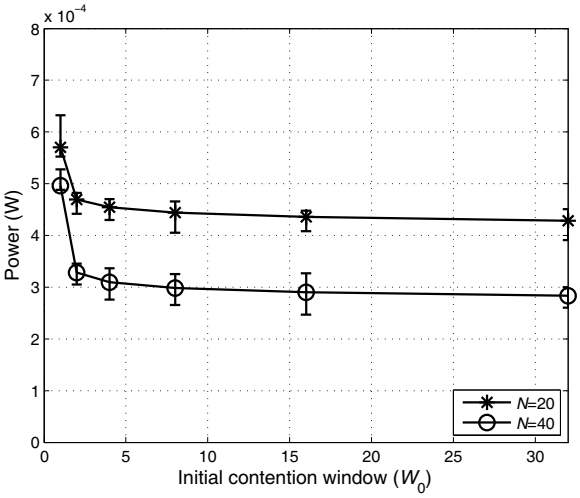


Fig. 15. Energy vs. initial contention window.

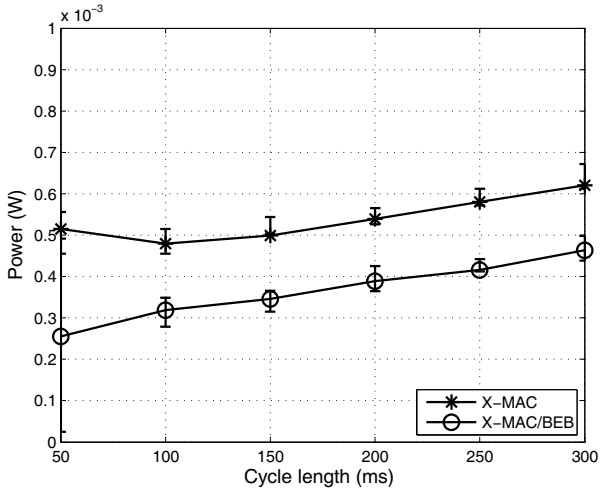


Fig. 14. Energy vs. cycle length.

VII. CONCLUSIONS AND FUTURE WORK

This paper introduces X-MAC/BEB, which incorporates a BEB mechanism in X-MAC to prevent the performance from rapidly deteriorating as WSNs become overcrowded. Along with X-MAC/BEB, this performance model is presented as an extension of X-MAC model. Both the analytical and simulation results are substantiated with less than a 3% deviation that X-MAC/BEB enhances the performance of X-MAC in term of throughput, latency, and energy efficiency especially when WSNs become populated. As the first future research topic, we will extend the X-MAC/BEB model to include the effect of channel errors [20] on the performance of X-MAC/BEB. We will also combine the Markov chain of the BEB algorithm with the X-MAC/BEB model to improve the accuracy of the X-MAC/BEB model for lightly loaded WSNs. Finally, we plan to evaluate the X-MAC/BEB algorithm over multi-hop mobile WSNs.

REFERENCES

[1] Y. Wei, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. IEEE INFOCOM*, June 2002, pp. 1567–1576.

[2] T. Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proc. ACM SenSys*, Nov. 2003, pp. 171–180.

[3] Shu Du, Amit Kumar Saha, and David B. Johnson, "RMAC: A routing-enhanced duty-cycle MAC protocol for wireless sensor networks," in *Proc. IEEE INFOCOM*, May. 2007, pp. 1478–1486.

[4] Yanjun Sun, Shu Du, Omer Gurewitz, and David B. Johnson, "DW-MAC: A low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks," in *Proc. ACM MobiHoc*, May 2008, pp. 53–62.

[5] W. Ye, F. Silva, and J. Heidemann, "Ultra-low duty cycle mac with scheduled channel polling," in *Proc. ACM SenSys*, Nov. 2006, pp. 321–333.

[6] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. ACM SenSys*, Nov. 2004, pp. 95–107.

[7] Amre El-Hoiydi and Jean-Dominique Decotignie, "WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks," in *Proc. ALGOSENSORS*, July 2004, pp. 18–31.

[8] M. Buettner, G. Yee, E. Anderson, and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proc. ACM SenSys*, Nov. 2006, pp. 307–320.

[9] Ashutosh Bhatia and R. C. Hansdah, "A distributed TDMA slot scheduling algorithm for spatially correlated contention in WSNs," in *Proc. IEEE AINA Workshops*, Mar. 2013, pp. 377–384.

[10] O. Yang and W. Heinzelman, "Modeling and performance analysis for duty-cycled MAC protocols with applications to S-MAC and X-MAC," *IEEE Trans. Mobile Comput.*, vol. 11, no. 6, pp. 905–921, June 2012.

[11] The Network Simulator, ns-2 [Online]. Available: http://www.isi.edu/nsnam/ns/

[12] G. Corbellini, E. C. Strinati, and A. Duda, "LA-MAC: Low-latency asynchronous MAC for wireless sensor networks," in *Proc. IEEE PIMRC*, Sept. 2012, pp. 380–386.

[13] A. B. Nacef, S. M. Senouci, Y. Ghamri-Doudane, and A. L. Beylot, "A cooperative low power mac protocol for WSNs," in *Proc. IEEE ICC*, June, 2011.

[14] S. H. Hong and H. K. Kim, "A multi-hop reservation method for end-to-end latency performance improvement in asynchronous MAC-based wireless sensor networks," *IEEE Trans. Consum. Electron.*, vol. 53, no. 3, pp. 1214–1220, Aug. 2009.

[15] J.-H. Lee, J.-W. Kim, and D.-S. Eom, "A delay-tolerant virtual tunnel scheme of asynchronous MAC protocol in WSN," *Wireless Pers. Commun.*, vol. 70, no. 2, pp. 657–675, May 2013.

[16] I. Park, H. Lee, S. Kang, and D.-S. Eom, "RIX-MAC: An energy-efficient receiver-initiated wakeup MAC protocol for WSNs," *Trans. Internet and Inf. Syst.*, vol. 8, no. 5, pp. 1604–1617, 2014.

[17] I. Park, J. Yi, and H. Lee, "A receiver-initiated MAC protocol for wireless sensor networks based on tree topology," *Int'l J. Distrib. Sensor Netw.*, 2015.

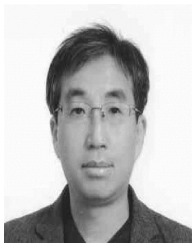
[18] N. Song, B. Kwak, J. Song, and L. E. Miller, "Enhancement of IEEE

802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm," in *Proc. IEEE VTC*, Apr. 2003, pp. 2775–2778.

[19] V. Vitsas, "Throughput analysis of linear backoff scheme in wireless LANS," in *Electron. Lett.*, vol. 39, no. 1, pp. 99–100, Jan. 2003.

[20] J. S. Ahn, J. H. Yoon, and K. W. Lee, "Performance and energy consumption analysis of 802.11 with FEC codes over wireless sensor networks," *IEEE J. Commun. Netw.*, vol. 9, no. 3, pp. 265–273, Sept. 2007.

Ayaz Ullah received M.S. degree in Computer Engineering from the Center for Advanced Studies in Engineering (CASE) Islamabad affiliated with the University of Engineering and Technology Taxila in 2009 and M.C.S. (Master Computer Science) degree from the Department of Computer Science Kohat University of Science and Technology (KUST) Kohat in 2006. He also received B.Sc. degree (Major in Computer Science) from the University of Peshawar in 2003. Currently, he is working toward the Ph.D. degree from the Department of Computer Science and Engineering, Dongguk University. His major interests are wireless sensor networks, IEEE 802.11, IEEE 802.15.4, and network simulation techniques.



Jong-Suk Ahn received the Ph.D. and M.S. degrees in Electrical Engineering from the University of Southern California in 1995 and from the Korean Advanced Institute of Science and Technology in 1985, respectively. He also received the B.S. degree from Seoul National University in 1983. He was a Visiting Researcher in ISI/USC for one year in 2001. He was awarded the Gaheon Prize for the best paper in 2003 from the Korean Information Science Society. He is currently a Professor at Dongguk University in Korea. His major interests are sensor networks, dynamic error control and flow control algorithms over wireless Internet, network simulation techniques, IEEE 802.11, and IEEE 802.15.4.