LER-MAC: A Load-Independent Energy-Efficient and Rate-Control Integrated Asynchronous Duty Cycle MAC for Wireless Sensor Networks

Muhammad Mostafa Monowar, Muhammad Mahbub Alam, Md. Obaidur Rahman, Choong Seon Hong
Department of Computer Engineering
Kyung Hee University
1 Seocheon, Giheung, Yongin, Gyeonggi, 449-701, South Korea
{monowar,mahbub,rupam}@networking.khu.ac.kr and cshong@khu.ac.kr

***Abstract*—Considering energy as a crucial resource, several duty cycle based MAC have already been proposed for Wireless Sensor Networks (WSNs) to gain higher energy efficiency during long idle period of the sensors, and are optimized for light traffic loads. Contrastively, considering the bandwidth constraint of WSNs, and to optimize the heavy traffic loads, another research trend is continuing in devising rate and congestion control at the MAC layer. To provide an integrated solution of both these distinct research trends at the MAC layer, in this paper, we present a *L*oad-independent *E*nergy-efficient and *R*ate-control Integrated Asynchronous Duty Cycle MAC (LER-MAC) for WSNs. Performance of LER-MAC has been evaluated using ns-2 which demonstrates that, LER-MAC conserves energy considerably during light traffic loads. Moreover, it procures higher throughput and lower latency avoiding packet drops through maximum utilization of the channel during heavy traffic loads.**

I. Introduction

The offbeat nature of the remote sensing applications (i.e., structural and habitat monitoring, fire detection, target recognition and tracking) for wireless sensor networks usually demand WSNs to alternate between long periods of low traffic volume (referred to as dormant state) and short periods of high traffic volume (referred to as crisis state)[1].

Considering energy as one of the most pivotal resources, the recent research trend in wireless sensor networks follows two distinct directions. One trend emphasizes in designing duty cycle based MAC with the aim of reducing the *idle listening* problem (unnecessarily turning on the radio while no channel activity is going on) [2],[3], taking into account the long dormant state of the nodes. While the other trend focuses on handling the high rate event traffic in the crisis situation adopting both the rate and congestion control at the MAC layer [4] [5] [6] [7]. These works intend to reduce the energy wastage due to packet drops, also aim to maximize the throughput which are overlooked by the current duty cycle based MAC protocols. Since, both these traffic situations are common for WSN applications and should be handled by a single MAC, an integrated solution at the MAC layer is thus essential, which to the best of our knowledge, remains unaddressed.

In this paper, we focus to design a duty cycle based MAC protocol for WSNs which aims to optimize the network irrespective of the traffic load exist in the network. In particular, during low traffic load, it tries to maximize the energy conservation, increasing the sleeping time as much as possible, which the network desires at that situation. On the other hand, during crisis state, it aims to maximize the throughput reducing the sleeping time, also reduce the packet loss, which the network demands during that period rather maintaining long duty cycle. However, because of the conflicting nature, meeting both these goals are challenging. Therefore, it is necessary to balance the energy conservation and network throughput based on the prevailing network traffic condition.

The state-of-the-art duty-cycle based MAC protocols can be classified as synchronous and asynchronous protocols based on their wake-up schedule maintenance for communication. In synchronous protocols [8] [9] [10] [11] nodes in a neighborhood wake up at the same time following a fixed wake up interval, and incur high cost for maintaining synchronization. Asynchronous protocols [12] [13] maintain independent wake up schedule and usually avoid this synchronization cost by employing preamble based transmission approach (data transmission is preceded by a long preamble). However, both these categories usually achieve a very low medium utilization due to high collision rate (synchronous protocols) and unnecessarily long medium occupancy due to preamble transmission (asynchronous protocols). Therefore, none of these protocols are suitable to handle high traffic situation. Nonetheless, a recent work proposes a receiver-initiated asynchronous duty-cycle MAC to minimize the control overhead on medium occupancy [14]. This work has motivated us to extend the duty-cycle based MAC to achieve a near optimal network throughput while keeping the nodes in sleeping states as maximum as possible.

Our major contributions include: i) We present LER-MAC, which integrated the objective of duty-cycle MAC with MAC based with rate/congestion control protocols using a unified mechanism. To the best of our knowledge, this is the first attempt which handles both the situation in a single mechanism. ii) We devise a receiver driven medium access mecha-

978-1-4244-7265-9/10/$26.00

nism which addresses higher energy efficiency, high medium utilization, and reduces the control overhead. iii) We introduce a novel bi-directional rate adjustment mechanism which exploits the receiver driven medium access technique. These mechanisms, in concert, achieve significant energy efficiency during dormant state than the existing asynchronous duty-cycle protocols, and also maximize the network throughput avoiding the packet drops during crisis state. iv) We perform extensive simulations using ns-2 to evaluate the performance of LER-MAC.

The remainder of the paper is organized as follows: Section II describes the LER-MAC protocol in detail, section III presents the performance evaluation using ns2, section IV summarizes the related works and finally, in section V, we present the concluding remarks with some future directions.

II. The LER-MAC Protocol

In this section, we begin by giving the overview of LER-MAC and then describe all of its components.

A. Overview

Figure 1 portrays the overview of our proposed scheme. The layered architecture is shown according to the 802.15.4 standard protocol stack for a sensor node. In this paper, we assume the nodes forward data to a sink in a multihop manner forming a tree structure. We refer to the nodes toward the sink as downstream nodes and the nodes toward the sources as upstream nodes. As in figure 1, LER-MAC is composed of two main components: *receiver-driven medium access* and *bi-directional rate adjustment*. The *receiver-driven medium access* is a contention based medium access technique in which data transmission is initiated upon receiving receiver's beacon. We apply multi-packet reception after each wake-up of a node which addresses both energy efficiency and high medium utilization. *Bi-directional rate adjustment* adjusts the duty-cycle/beacon sending rate, based on prevailing traffic condition, employing *forward rate adjustment* and *backward rate control* mechanisms. *Forward rate adjustment* adjusts the beacon sending rate of a node based on the request from its upstream senders. It further comprises three operations. First, the *multiple packet reception adjustment* determines the number of packet reception per wake-up taking into account the current traffic load at the upstream senders. Second, the *traffic-rate aware wake-up interval selection* selects the wake-up/beacon interval of a node considering the data generation or wake-up interval of the upstream nodes to address the traffic-load variation based duty-cycle adjustment. Third, the *rate-adjustment avoiding queue build-up* determines the appropriate beacon sending rate with the aim of avoiding queue build-up while the traffic load at the upstream senders becomes higher. On the other hand, *backward rate control* controls the forwarding rate of the nodes considering the network status prevailing at their corresponding downstream node, employing two operations. First, the *rate control addressing high contention* controls the beacon sending rate addressing the high contention around a node and second, *source rate control addressing congestion* controls the source rate while congestion occurs through creating back-pressure. We perform both upward and downward cross layer information flow between MAC and Application layer to pass the congestion detection information for rate control and to determine the wake-up interval based on the sampling rate of the sources respectively.

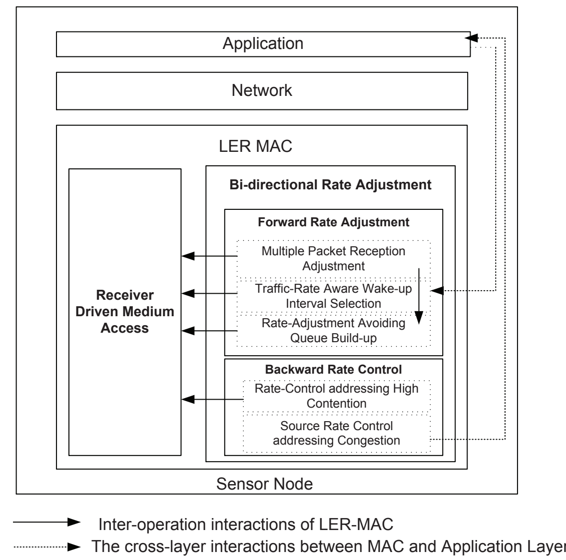


Fig. 1. Overview of LER-MAC

B. Receiver-driven medium access

The basic mechanism of receiver driven medium access is similar to RI-MAC. However, our proposed medium access mechanism enhances RI-MAC in terms of energy efficiency and high medium utilization employing multiple packet reception, suitable sleeping time determination, control packet reduction and different back-off strategy. We illustrate this mechanism in figure 2 and describe it according to the different roles of the nodes as follows.

Node functioning as a receiver. Following a periodic wake-up interval, a node wakes up to receive packets from its upstream senders. Upon activating the radio, a node performs a short beacon backoff (BO_b) to avoid the collision from simultaneous beacon packet transmissions by the neighboring nodes and then broadcasts a beacon (referred to as primary beacon) after CCA check. It then waits to receive packets from any of its upstream senders until a time-out (TO) occurs. The receiver turns into sleeping state if it detects the channel busy during beacon transmission or detects collision during waiting for packets or no packets arrived after TO occurs. A receiver could shift its wake up time on experiencing consecutive channel busy to avoid the simultaneously ongoing transmission(s) with its reception time.

A node receives multiple packets at each reception round (which starts after transmitting the beacon) for high medium utilization. The number of packets to be received at a particular reception round is denoted as N_r and it is embedded in the

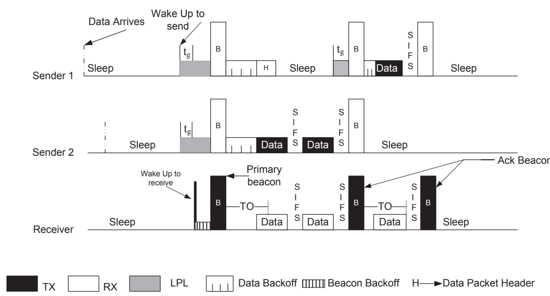


Fig. 2. The value of N_r is set as 3 as for example. Sender 2 wins the channel but have only 2 packets in the queue. It goes to sleep after transmitting all the packets in its queue. Sender 1 overhears the first packet header of sender 2 and then goes to sleep. It wakes up again with a guard time and sends the remaining number of packets in that reception round and goes to sleep again.

beacon packet by the receiver. The receiver acknowledges the successful reception of packets of a particular sender through sending an ACK-Beacon which acts as Block-ACK. However, if the receiver receives less number of packets than N_r, it updates the remaining number of packets to be received in the ACK-Beacon packet, which invites other potential senders having data to send packets to utilize that reception round. A receiver goes to sleep after full utilization of the reception round. The role of ACK-Beacon is thus twofold; first, it acts as acknowledgement, second it solicits for data packets to utilize the reception round.

Node functioning as a sender. A node having data in the queue acts as a sender. To avoid the idle listening as is suffered by the RI-MAC sender, every sender is informed of its receiver's wake-up schedule. For this, the receiver i appends the remaining duration until its next wake-up, t_r^i, in the primary beacon or ACK-Beacon which is calculated as-

$$t_r^i = d_w^i - (t_b^i - t_w^i) \tag{1}$$

where, d_w^i is the wake-up interval of node i, t_b^i is the primary or ACK-beacon sending time and t_w^i is the time at which node i woke up. Using the value t_r^i, a sender j determines its wake-up schedule for sending data, $t_w^j(send)$ as-

$$t_w^j(send) = t_c^j + (t_r^i - t_g) \tag{2}$$

where, t_c^j is the current time at sender j, t_g is the guard time for clock drift and is measured as- $t_g = 2 \times R_{drift} \times t_r^i$; where, R_{drift} is the maximum clock drift rate.

A sender waits in LPL for receiver's beacon after wake up to send data. Upon detecting the beacon preamble it powers up and receives the entire beacon packet. However, if a sender misses the sending operation in its pre-scheduled sending time due to receiver's beacon failure or collision or if it is a joining node, it continues waiting in LPL for receiver's beacon instead of sleeping.

The senders start their random back-off period within a fixed contention window, BO_d after receiving the beacon. A wining node j sends N_s^j packets in a succession with a gap of SIFS within each packet. The value of N_s^j is set as the existing queue length, q_c, if $q_c \leq N_r^i$, otherwise it is set as the value of N_r^i.

The wining sender goes to sleep immediately after ACK-Beacon reception. The losing node's behavior depends on two cases. *First*, the losing nodes will be in sleep state if they find $N_s^j = N_r^i$ through overhearing of the first data packet header (which contains the value of N_s^j). *Second,* for the case, $N_s^j < N_r^i$, the losing nodes remain awaken instead of sleeping if $N_s^j < threshold$, to reduce the energy cost caused by frequent turning on/off the radio. We set the threshold as 2. If $N_s^j \geq threshold$, the losing nodes go to sleep. They wake up again just before the ACK-Beacon reception time (intended for wining sender) with a guard time t_g, and wait for it in LPL. The nodes can easily estimate the sleeping duration before ACK-Beacon comes through overhearing the value, N_s^j from the packet header. In every case, whenever a node looses the channel it freezes its backoff to get higher priority in accessing the channel for the next contention which signifies the fair access of the medium.

C. Bi-directional rate adjustment

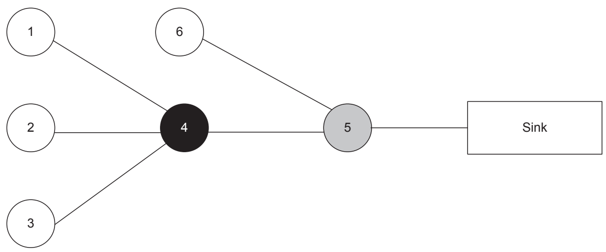
To come up with the design motivations of LER-MAC, that is, maximizing the energy saving during low traffic load and gaining the near optimal throughput during high traffic rate situation, we contrive bi-directional rate adjustment mechanism which comprises *forward rate adjustment* and *backward rate control* mechanisms. It is to be mentioned that, in this protocol, controlling duty cycle is analogous to the control of its beacon interval (or beacon rate) and vice-versa since a node broadcasts primary beacon at each wake up interval. Hence, throughout the paper, the term wake-up interval and beacon interval have been used interchangeably.

1) Forward rate adjustment (FRA): In this mechanism packet receiving rate of a node is controlled according to the request performed by the upstream nodes. It includes the following operations.

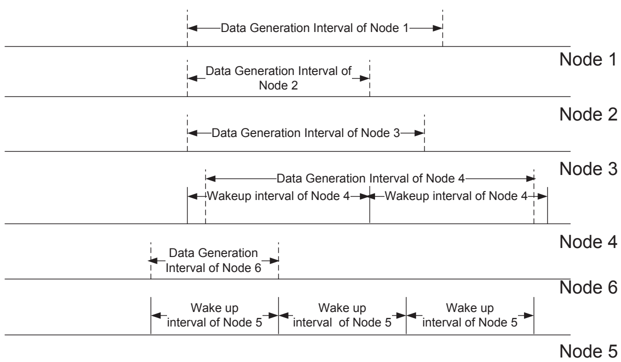
Multiple packet reception adjustment. Since, the proposed receiver driven medium access mechanism allows multiple packet reception at each wake-up interval, this value also affects the forwarding rate of the upstream nodes. As the traffic load at upstream nodes varies dynamically, thus the adjustment of multiple packet reception should reflect accordingly. However, a maximum limit of this value should be provided to address both higher medium access delay and collision, denoted as N_r^{max}. As long as the load at the upstream senders persists within this limit, a receiver could receive all the packets for higher medium utilization.

In this operation, a node, i sets the initial value of N_r as N_r^{max} to force the upstream senders to transmit all the queued packets in order to get an idea of the current load. To let the receiver know about the observed load at upstream senders, the senders include their corresponding current load in terms of packet rate, L_c in the packet header which is calculated as-

$$L_c = PacketReceptionRate + PacketGenerationRate$$
$$= \frac{N_r}{d_w} + \frac{1}{T_d} \tag{3}$$



(a) An example data gathering tree. White nodes stand for source only, grey as forwarding only and black acts both as source and forwarding node.



(b) Wake-up interval of node 4 and node 5. Node 4 updates it wake-up interval as equal as data generation interval of node 2 (the minimum interval among node 1, 2, 3 and 4). Similarly node 5 updates its wake-up interval as node 6's data generation interval.

Fig. 3. Traffic-rate aware wake-up interval selection.

where d_w is the wake-up interval of a node and T_d is the data generation interval. A receiver i then estimates the observed load per wake up at its upstream senders as-

$$OL^i = \{ \sum_{j \in S(i)} L_c^j \} \times d_w^i \tag{4}$$

where, $S(i)$ is the set of upstream senders of node i and d_w^i is the wake up interval of node i. Node i sets the value of N_r as N_r^{max}, when the observed load exceeds or is equal to N_r^{max} otherwise, it is set as the value of current observed load, OL^i.

Traffic-rate aware wake-up interval selection. Traffic-rate aware wake-up interval selection (TWIS) determines the wake-up/beacon interval of downstream nodes based on upstream node's request. To reflect the current traffic rate at upstream senders, the wake-up interval of a node i, d_w^i is calculated as-

$$d_w^i = \min(d_w^j); j \in S(i) \tag{5}$$

TWIS demands the upstream senders to append data generation interval (if it is source node only) or wake-up interval (if it is forwarding node only) or minimum of the packet generation interval and wake-up interval (if it acts both as source and forwarding node) in the interval field of data packet header while sending packets to the receiver. Thus, the wake-up interval is selected according to the minimum wake-up interval or data generation interval of the upstream senders according to Eq. 5. However, initial wake-up interval could be chosen according to maximum data generation interval (decided by the application) or any random value and eventually update it according to this operation.

Figure 3 illustrates the TWIS operation. TWIS obtains the maximum energy efficiency during low traffic load, since nodes perform long sleep operation during dormant state. TWIS also avoids unnecessary wake up since, at each wake up, nodes expect to receive a packet from any of its upstream senders. Besides, during the crisis state, nodes wake up at a high rate which also increases the beacon sending rate to achieve higher throughput. This also reduces the per-hop latency, and for multi-hop network TWIS affirms the non-increasing wake-up interval selection by the downstream nodes to maintain lower end-to-end delay.

Rate adjustment avoiding queue build up. The stated TWIS operation prevents queue build-up at the upstream nodes while the observed load is in tolerable situation ($OL^i \leq N_r^{max}$). However, when $OL^i > N_r^{max}$, to avoid queue build up at the upstream senders, and to accommodate the extra traffic prevailing at its upstream senders, a node i estimates its required beacon sending rate, $R_{BS}^i(req)$ as-

$$R_{BS}^i(req) = \frac{R_{BS}^i(c) \times OL^i}{N_r^{max}} \tag{6}$$

In Eq. 6, $R_{BS}^i(c)$ is the current beacon sending rate of node i and OL^i is the current observed load. Based on Eq. 6 a node sets its new beacon sending rate, $R_{BS}^i(new)$ as-

$$R_{BS}^i(new) = R_{BS}^i(c) + \alpha(R_{BS}^i(req) - R_{BS}^i(c)) \tag{7}$$

Here, α is the stabilizing parameter. If $\alpha = 0$ then it nullifies the effect of required rate. Again, if $\alpha = 1$, then $R_{BS}^i(new) = R_{BS}^i(req)$; which could result a sharp increase of the beacon sending rate instantaneously and that might create an oscillating situation. Therefore, the value should be, $0 < \alpha < 1$ and empirically we set the value as 0.35.

As the new rate approaches to required rate eventually, the beacon sending rate of a node increases with the aim of gaining highest possible throughput during crisis state avoiding queue build up. This situation also allows the nodes to sleep though for short duration. However, to provide energy efficiency, a node might remain awaken instead of sleeping if $E_l^{d_w} \leq E_{onoff}$ holds true; where, $E_l^{d_w}$ is the energy consumption for listening up to the wake-up interval, and E_{onoff} denotes the energy consumption for turning the radio on and off.

2) Backward rate control (BRC): Disparate the forward rate adjustment, the backward rate control operation controls the beacon sending rate as well as source rate, considering the current network status observed at the node itself. The related operations of this mechanism include:

Rate control addressing high contention. The high contention around a node prevents to utilize the beacon rate measured through forward rate adjustment mechanism. Hence, this is not the optimal wake-up/beacon interval considering the contention scenario. Therefore, a node adjusts its beacon interval during high contention. Algorithm 1 exhibits the rate adjustment during high contention. In this operation, every node maintains a contention resolution counter (CRC) and observes the number of failed transmission, N_f, (a node experiencing collision after sending beacon in a reception round is referred to as failed transmission) within this counter value (line 1-9 of Algorithm 1). During the observation, if the beacon interval changes, a node reinitializes its CRC value. After the observation period (when CRC reaches to 0), if the value of N_f becomes greater than a threshold, μ (80% of CRC), it immediately performs an additive decrease of beacon sending rate (line 11-12 in Algorithm 1). Additive decrease is used to avoid the aggressive rate reduction which could be caused by multiplicative decrease mechanism. However, while the value of N_f is less than μ and prolong for m consecutive observation period (line 17-20 of Algorithm 1), a node updates its beacon interval according to forward rate adjustment mechanism.

Algorithm 1 Beacon rate control addressing high contention

```
1.  Initialize CRC
2.  while (CRC ≥ 0) do
3.      if R^i_BS(current)changes then
4.          go to 1
5.      end if
6.      if Experience Collision then
7.          N_f ++
8.          CRC − −
9.      end if
10. end while
11. if (N_f ≥ threshold) then
12.     R^i_BS(new) = R^i_BS(c) − γ
13.     R^i_BS(c) = R^i_BS(new)
14.     d^i_w(c) = 1 / R^i_BS(c)
15.     j = 0
16. end if
17. if (N_f < threshold and j ≤ m) then
18.     j ++
19.     if (j==m) then
20.         call FRA()
21.     end if
22. end if
```

Source Rate Control addressing Congestion. The reduced packet reception rate due to high contention compared to the rate measured by the forward rate adjustment mechanism and limited capacity of wireless network could cause queue build up at a node which in turn causes packet loss due to queue overflow. Thus, source rates need to be decreased which in turn increases the beacon/wake-up interval of the nodes along the path to the sources.

For this, we measure the instantaneous average queue length of a node to detect the level of congestion following the usual trend of congestion control mechanisms. [4] [5][15]. The average is taken using the following equation-

$$Avg_q^i = (1 - \varepsilon) \times Avg_q^i + \varepsilon \times inst_q \qquad (8)$$

Where, Avg_q^i is the average queue length of node i, $inst_q^i$ is the instantaneous queue length of that node and ε is the tuning parameter. The average queue length is updated after each reception round of node i.

Whenever the value of Avg_q^i exceeds a certain threshold, λ, a node sets the ECN bit in the primary beacon and ACK-Beacon packet which is propagated by the nodes along the path to the sources through their corresponding beacon packets to notify the source nodes regarding congestion. A node resets the ECN bit upon relinquish of congestion. Sources employ the commonly followed Additive Increase Multiplicative Decrease (AIMD) approach for rate control. Since, all the sources along the path to the congested node reduce their rate and increases at a same rate afterwards; thus fairness is maintained among the sources [16].

This source rate control mechanism also might not avoid the packet drops fully if the sources are far away from the congested region. Hence, to fully avoid the packet drops, a receiver i could stop sending beacon if it detects the queue is full. It resumes sending the beacon again when its queue has space to receive at least one packet. In this case, the value N_r is set as the remaining space of its queue.

D. Parameter selection in LER-MAC

LER-MAC uses several parameters in different operations. This section focuses on the details of the choice of parameters as used in LER-MAC.

Choice of N_r^{max}. The value of N_r^{max} is a system parameter and completely depends on the topology and application requirement. We consider the channel busy probability to select this value which also reflects the medium access delay and collision. We derive the maximum value of N_r^{max} based on the analysis performed in [17] and [18] for IEEE 802.15.4 CSMA/CA mechanism. From their results obtained, we formulate the channel busy probability, denoted as β, due to the transmission from neighboring nodes, as-

$$\beta = \frac{n \times T_t^{avg}}{d_w - T_t^{avg}} \qquad (9)$$

Where, T_t^{avg} is the average transmission time required to transmit N_r packets which can be obtained as-

$$T_t^{avg} = N_r \times L_d T_b + (N_r - 1) \times SIFS + 2 L_b T_b \qquad (10)$$

We use the parameter values for L_d, $SIFS$, L_b, T_b as used in simulation discussed in section III. Assuming 200ms[1] as d_w, keeping the channel busy probability as 75% and putting the average number of backlogged neighbors as 5 (depends on the topology), and solving the Eq. 9 using Eq. 10, we obtain, $N_r^{max} = 3$ and use that value in the simulation.

[1]The value could be taken from the maximum allowable data generation rate or minimum allowable data generation interval decided by the application.

Choice of congestion control parameters. We intend to keep the value of congestion detection threshold, λ (as discussed in section II-C2) lower to detect an early congestion and initiate the source rate control immediately for congestion avoidance. However, a very low threshold might result lower network utilization and a very high value also might delay in detecting congestion. From the simulation, we obtain that 60-70% of the maximum queue size gives better result for early handling of congestion. It leads to set the value of λ as 9. Hence, when the average queue length reaches 9 the source rate control mechanism is initiated.

The determination of the value of ε used in Eq. 8 should reflect both the transient congestion situation and actual queue size. If it is too large, it could overestimate the congestion situation and a very lower value also might not reflect the actual queue size. The relationship between congestion detection threshold and tuning parameter has been explored in [19]. Based on their analysis and through extensive simulation we set the value of ε as 0.1. This value detects congestion when the actual queue size reaches 63% of maximum queue length (40 as set in the simulation).

Choice of other parameters. We used the parameters CRC and m in algorithm 1. Both these parameters signify the observation period to determine the contention level around a node. We intuitively set the value of CRC and m in the simulation which justifies the selection.

III. Performance Evaluations

In this section, the performance evaluation of LER-MAC has been presented based on different parameters and the results exhibit its effectiveness.

A. Simulation Environment

We evaluate our protocol through extensive simulation using ns-2 network simulator. Ns-2 version 2.33 has been used with the Two Ray Ground propagation model in the air and a single Omni-directional antenna. A network of area 100m x 100m is used with 100 nodes deployment in uniform random distribution. We evaluated our protocol using two models with this network environment: Correlated Event Traffic model and Periodic Traffic Load model. In this study, we compare LER-MAC with RI-MAC and a IEEE 802.11 like always on protocol. IEEE 802.11 has been chosen as it achieves the highest possible throughput since packet transmission is initiated as soon as it arrives to the queue and no duty cycle is maintained. We further omit preamble based asynchronous protocols as RI-MAC showed its supremacy upon those protocols in their simulation. To simplify our evaluation, we exclude routing traffic and assume that a routing functionality exist which selects the shortest path between any two nodes. Table I represents different parameters used in our simulation. The power consumption values and some of the parameters (SIFS, Slot time, Data Rate, CCA Check delay) are from the data sheet of CC2420 radio [20]. The beacon size varies from minimum 96 to maximum 113 bits depending on the Block Ack size along with the presence/absence of congestion

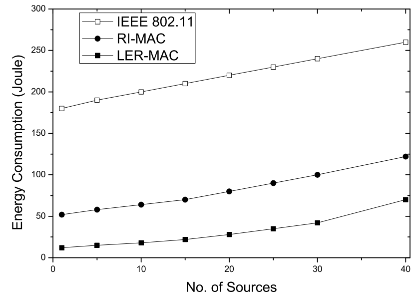


Fig. 4. Average Energy Consumption varying number of sources.

control information. The sleep interval of RI-MAC is set to a random value between 0.5 to 1.5 second. Each simulation has been performed for 60 seconds and we averaged the value obtained for 30 random runs.

B. Simulation Results

1) Correlated Event Traffic Model: This model captures the effect of spatially correlated events in a sensor network. The remote sensing applications generate high volume of traffic during the occurrence of an event and both the number of sources and the traffic generation rate affect the traffic load situation. To reflect this scenario, this model picks a random location for the generation of event. The event radius is varied from 6 m to 36 m and thereby got the number of sources[2] from 1 to 40. Moreover, we assume that an event occurred after the 10 seconds from the beginning of the simulation and this crisis situation lasts for 20 seconds. During the crisis state, the data generation interval for the sources is set as 0.05 second whereas in the dormant state it is set as 10 second (assuming it as maximum data generation interval). In this study, the traffic load is varied changing the number of sources.

Figure 4 depicts the total energy consumption among the sensor nodes with increasing traffic load. IEEE 802.11 exhibits the highest energy consumption due to "always on" feature and idle listening dominates the energy consumption. Due to duty cycle maintenance, both LER-MAC and RI-MAC show significantly lower energy consumption than IEEE 802.11; although LER-MAC conserves energy far better than RI-MAC, because of its load aware duty cycle maintenance using bi-directional rate control. Hence, during dormant period, nodes in LER-MAC stay in sleeping state much longer than RI-MAC. Moreover, during high traffic load, although the wake-up interval of LER-MAC decreases with the increase of packet sending rate; but the receivers still perform sleep operation after receiving N_r^{max} packets and senders are also tuned to the receiver's wake-up time thus avoiding idle listening.

The average end-to-end latency per packet during crisis state with increasing traffic load is shown in Figure 5. IEEE 802.11 shows the best performance for latency since it is exempted from any sleep delay or beacon waiting delay for data transmission. However, as the traffic load increases, the

[2]No of sources= $\rho\pi R_s^2$ where, $\rho = \frac{100}{100\times100}, \pi = 3.14, R_s = sensing\ range$

TABLE I
SIMULATION PARAMETERS FOR LER-MAC

Parameter	Value	Parameter	Value	Parameter	Value
Data Rate	250 kbps	Beacon length	$96-113$ bits	N_r^{max}	3 packets
SIFS	$192\mu s$	MAC Header	14 bytes	λ	9 packets
Slot Time	$320\mu s$	Payload Size	32 bytes	ε	0.1
Tx Range	30 m	BO_b	8	CRC	5
Carrier Sensing Range	67 m	BO_d	32	μ	4
CCA Check Delay	$128\mu s$	Queue Size	40 packets	m	3
PHY Header	6 bytes	Retry Limit	5	γ	1
Listening Power	$56.4mW$	Transmitting Power	$52.2mW$	Receiving Power	$56.4mW$
Polling Power	$12.3mW$	Sleeping Power	$3\mu W$	Timeout	$10.5ms$

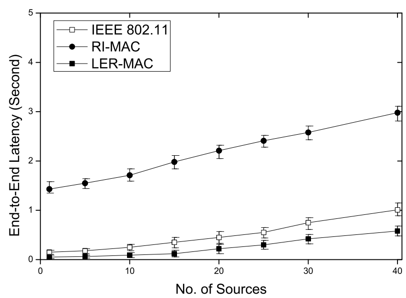


Fig. 5. Average End-to-End latency varying number of sources

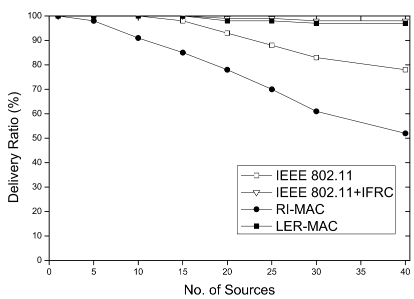


Fig. 6. Delivery Ratio varying Number of Sources

end-to-end latency per packet increases for all the protocols due to the retransmission of data packet caused by collisions. LER-MAC exhibits the closest performance with IEEE 802.11 due to the bi-directional rate adjustment which increases the beacon rate as maximum as possible addressing both the contention and congestion situation although it allows some sleep delay. RI-MAC shows the worst case latency due to non-adaptive duty cycle maintenance along with higher collision during crisis state.

Figure 6 depicts the delivery ratio performance of the protocols with different traffic load. In this study, we implement a MAC based distributed rate control mechanism (IFRC) with IEEE 802.11 to know the behavior of it with a rate control mechanism. Considering the delivery ratio, both LER-MAC and IEEE 802.11 with IFRC achieves almost 100% delivery ratio due to the rate control techniques. It is worth mentioning that, this is our desired goal, that is, to achieve the higher delivery ratio as close as 802.11 like always on protocol having rate control mechanism during crisis state.

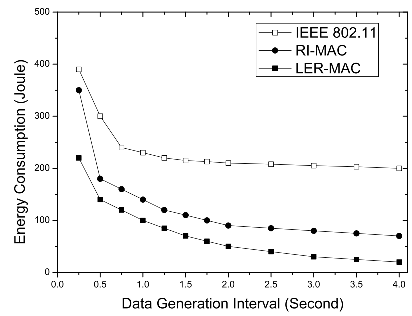


Fig. 7. Energy Consumption varying Data Generation Interval

However, the absence of rate control mechanism inhibits to attain a higher delivery ratio for both IEEE 802.11 without rate control and RI-MAC notwithstanding, RI-MAC shows the worst performance due to huge number of collision and congestion loss for maintaining non-adaptive duty cycle during crisis state.

2) Periodic Traffic Load Model: This model represents the periodic application (i.e. monitoring application) in which sensors generate data at a fixed interval. In this model, each sensor sources traffic at a particular offered load as well as forward other nodes traffic through multi-hop manner. Here, we randomize the initial data generation of the sensors to avoid the synchronized periodic reports of the sensors. In this study, the traffic load is varied by changing the data generation interval of the nodes.

Figure 7 illustrates the impact of data generation interval on energy consumption. As the figure shows, high rate periodic applications consume significantly much energy than the applications with moderate traffic rate. For the high rate traffic, energy depletion for transmission and reception is more dominating than idle listening. However, as the traffic rate decreases, idle listening dominates the energy conservation. RI-MAC and LER-MAC both show considerable energy saving than IEEE 802.11 due to the handling of idle listening. In this regard, LER-MAC delineates better performance than RI-MAC as it reduces the idle listening significantly due to its traffic rate aware wake up scheduling as well as avoiding of energy wastage due to packet drops.

Figure 8 shows the effect of data generation interval on average end-to-end latency per packet. While the data gen-

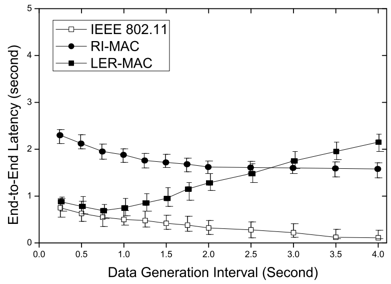
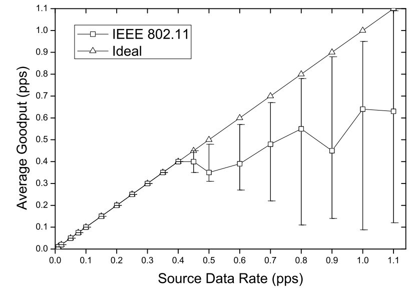


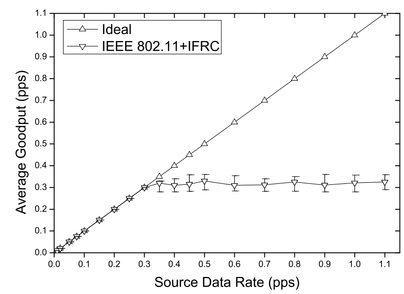
Fig. 8. Average End-to-End Latency varying Data Generation Interval

eration rate is high, LER-MAC performs well with IEEE 802.11 having lower latency. However, in this period, RI-MAC incurs high latency due to the retransmission per packet increases for collisions. As the data generation rate decreases, LER-MAC also shows higher end-to-end latency as the sleep delay increases. This reflects the design objective of LER-MAC, in particular, achieving lower latency during high traffic rate, while during low traffic rate, gaining the higher energy efficiency. Thus, as far as the latency is considered, LER-MAC is well suited for the periodic applications that generate high rate traffic which is quite common in practice for WSNs. (i.e., structural monitoring, habitat Monitoring).

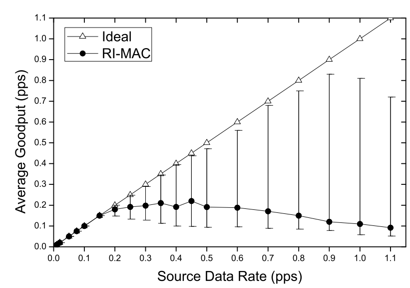
Figure 9 depicts the average goodput achieved over all nodes varying source data rate for different protocols. For each case, the diagonal line represents the achievable rate with infinite capacity also termed as ideal case. The error bars parallel to the y-axis indicates the maximum variation in node goodput at each offered load. As the Figure 9(a) shows, the maximum achievable fair rate for IEEE 802.11 with no rate control mechanism is 0.42 pps, after which, although the average goodput increases, but the variability in goodput also increases. The high variation in error bars indicates the low fairness achieved among the nodes. In particular, nodes closer to the sink achieve higher goodput than the nodes which are far away. While IFRC is used, IEEE 802.11 shows fair per node goodput as shown in Figure 9(b), although the maximum achievable goodput decreases (0.3 pps). LER-MAC also demonstrates fair goodput per node (Figure 9(d)). However the maximum achievable goodput for LER-MAC (0.38 pps) is better than IEEE 802.11 with IFRC because of bi-directional rate control mechanisms which controls the rate less aggressively than IFRC does, considering both contention and congestion. On the contrary, the maximum sustainable fair rate for RI-MAC is only 0.15 pps (Figure 9(c)) because of the non-adaptive duty cycle maintenance as traffic load varies, and it lacks rate control mechanisms.



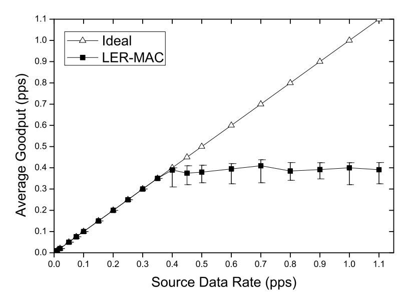
(a) Average Goodput for IEEE 802.11



(b) Average Goodput for IEEE 802.11 with IFRC



(c) Average Goodput for RI-MAC



(d) Average Goodput for LER-MAC

Fig. 9. Average Goodput achieved varying Source data rate

IV. Related Work

In recent years, energy-efficient duty cycle MAC protocols have been a very prominent research area in WSNs. Among the asynchronous protocols, B-MAC [12], is a contention based CSMA like approach which uses long preamble before data transmission that last longer than receiver's wake-up interval to ensure the reception of data by the receiver. B-MAC exhibits higher energy efficiency in low traffic loads since after each wake-up it performs very short channel activity checking. However, it suffers overhearing problem and shows higher latency for long preamble transmission.

X-MAC[13], is the optimized version of B-MAC to overcome the overhearing problem through introducing short strobed preambles and inclusion of target receiver address in the preamble. This facilitates in sending early acknowledgement to the sender thereby conserving energy as well as reducing per-hop latency. Yet, at each wake-up every node has to perform long CCA check in LPL, (longer than ACK waiting period) which causes energy wastage if no data available at the sender. Moreover, X-MAC also occupy the channel for a long time with preamble transmission.

RI-MAC[14] handles the problems of long channel occupancy by the senders existed in B-MAC and X-MAC introducing receiver-initiated transmission in which data transmission is initiated through the transmission of beacon by the receiver. RI-MAC achieves better energy efficiency, delivery ratio and latency compared to X-MAC. However, like the prior asynchronous protocols, RI-MAC also demonstrates non-adaptive behavior in maintaining wake-up interval and the channel utilization performance of RI-MAC during extreme conditions is still unknown.

The state-of-the-art congestion/rate control protocols for WSNs achieve energy efficiency through reducing the energy wastage caused by packet drops by imposing sustainable rate to the sources. These protocols utilizes the maximum channel capacity for gaining higher throughput. Due to the dependency of MAC layer information for congestion control, MAC based distributed congestion control protocols are developed such as CODA [5], FUSION [15], IFRC [4], CCF [7], PCCP [6] etc and are more effective in wireless sensor network environment. Nonetheless, these protocols are developed based on the IEEE 802.11 like "Always on" MAC protocol, and the effect of duty-cycling on congestion/rate control during crisis state has not been studied so far.

V. Conclusions and Future Work

This paper presents LER-MAC, which integrates the rate control functionalities with asynchronous duty cycle medium access control mechanism. The simulation results obtained using ns-2 reveals that LER-MAC well balances among energy conservation, throughput maximization and delay minimization irrespective of the traffic load in the network.

The design of LER-MAC points to some directions for future work: an analytical validation of LER-MAC, more detailed evaluation with other metrics and implementing on real test-bed scenario.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by MEST (No. 2009-0083838) Dr. CS Hong is the corresponding author.

References

[1] J. Kang, Y. Zhang, and B. Nath, "Tara: Topology-aware resource adaptation to alleviate congestion in sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 7, pp. 919–931, 2007.

[2] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 7, no. 3, pp. 537–568, 2009.

[3] G. P. Halkes, T. van Dam, and K. G. Langendoen, "Comparing energy-saving mac protocols for wireless sensor networks," *Mob. Netw. Appl.*, vol. 10, no. 5, pp. 783–791, 2005.

[4] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis, "Interference-aware fair rate control in wireless sensor networks," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 63–74, 2006.

[5] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "Coda: congestion detection and avoidance in sensor networks," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2003, pp. 266–279.

[6] C. Wang, B. Li, K. Sohraby, M. Daneshmand, and Y. Hu, "Upstream congestion control in wireless sensor networks through cross-layer optimization," *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 4, pp. 786–795, May 2007.

[7] C. T. Ee and R. Bajcsy, "Congestion control and fairness for many-to-one routing in sensor networks," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2004, pp. 148–161.

[8] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, 2004.

[9] S. Du, A. Saha, and D. Johnson, "Rmac: A routing-enhanced duty-cycle mac protocol for wireless sensor networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, May 2007, pp. 1478–1486.

[10] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency mac for tree-based data gathering in sensor networks: Research articles," *Wirel. Commun. Mob. Comput.*, vol. 7, no. 7, pp. 863–875, 2007.

[11] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, "Dw-mac: a low latency, energy efficient demand-wakeup mac protocol for wireless sensor networks," in *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2008, pp. 53–62.

[12] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2004, pp. 95–107.

[13] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks," in *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2006, pp. 307–320.

[14] Y. Sun, O. Gurewitz, and D. B. Johnson, "Ri-mac: a receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks," in *SenSys '08: Proceedings of the 6th ACM conference on Embedded network sensor systems*. New York, NY, USA: ACM, 2008, pp. 1–14.

[15] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating congestion in wireless sensor networks," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2004, pp. 134–147.

[16] Y. Jian and S. Chen, "Can csma/ca networks be made fair?" in *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2008, pp. 235–246.

[17] T. O. Kim, J. S. Park, H. J. Chong, K. J. Kim, and B. D. Choi, "Performance analysis of ieee 802.15.4 non-beacon mode with the unslotted csma/ca," *Communications Letters, IEEE*, vol. 12, no. 4, pp. 238–240, April 2008.

[18] J. Na, S. Lim, and C.-K. Kim, "Dual wake-up low power listening for duty cycled wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2008, p. 11, 2008.

[19] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, 1993.

[20] "Chipcon inc. CC2420 datasheet." http://www.chipcon.com.