# LEB-MAC : Load and Energy Balancing MAC Protocol for Energy Harvesting Powered Wireless Sensor Networks

Huey-Ing Liu, Wen-Jing He
Department of Electrical Engineering
Fu-Jen Catholic University, Taiwan, ROC
{hiliu, 499216325}@mail.fju.edu.tw

Winston KG Seah
School of Engineering & Computer Science
Victoria University of Wellington, New Zealand
winston.seah@ecs.vuw.ac.nz

***Ambient energy from solar, vibration, heat and wind provide alternative energy sources to power sensors and extend the lifetime of wireless sensor networks which have traditionally been powered by batteries. This paper aims to enhance the performance of energy harvesting powered wireless sensor networks in three aspects: relaying, scheduling, and medium access control. To better adapt to the characteristics of energy harvesting, an asynchronous receiver-initiated duty-cycling approach is preferred in energy harvesting powered wireless sensor networks. This reduces the duty cycle of senders, and regulates the active and sleep intervals according to the energy levels of sensors. When nodes run out of power and need time to recharge, network holes or voids develop, forcing data packets to be routed via other paths, like detours. The proposed relaying strategy aims to prevent holes by balancing the load across the network according to nodes' energy harvesting characteristics. This is a natural consequence of the asynchronous duty cycling by scheduling transmission based on the receiver's availability. The simulation results show that our scheme outperforms in terms of sender duty cycle, end-to-end delay and delivery ratio, especially in challenged conditions where other protocols fail.***

***Keywords—Energy harvesting, MAC, Wireless sensor network***

## I. Introduction

Wireless sensor networks (WSNs) have traditionally been powered by non-renewable sources, e.g. batteries, which limited their operational lifetimes. This has motivated extensive research on how to prolong the battery lifetime using energy efficient schemes. However, the emergence of new WSN applications like structural health monitoring, rare events monitoring and in-situ environmental monitoring have pushed the operational duration of WSNs far beyond what batteries can provide. Worse, battery replacement is difficult, if not impossible, and alternative means to power them are needed. The availability of different natural sources of ambient energy make energy harvesting (or scavenging) a viable alternative for extending the operational lifetime of wireless sensor networks, giving rise to energy harvesting (EH) powered WSNs or EH-WSNs [1][2].

However, current EH technology that can be integrated into wireless sensors can only provide intermittent energy – a trend that continues to prevail despite rapid advances in EH technologies. It has been shown that fundamental mechanisms in widely used wireless medium access control (MAC) protocols, like backoff and retransmission, are very likely to be non-optimal [3]. This is due to the unpredictable availability of harvested energy, that makes timing schedules and duty cycles hard to enforce when the amount of energy that can be harvested cannot be determined a priori. While duty-cycling was effective for alleviating network congestion and node contention in traditional WSNs [4], where nodes' remaining energy can be easily measured, such actions are detrimental in EH-WSNs as waiting consumes energy and also energy is lost via capacitor leakage even when a node is not operating [5]. This results in intermittent connectivity and unpredictable network topology changes.

In order not to waste energy transmitting when there is no receiving node awake, we use a receiver-initiated transmission approach where a (receiver) node broadcasts a beacon to notify its neighbours that it is ready to receive a packet. Based on its past energy harvesting history, the node can also compute and broadcast the next time it will wake up with a certain level of probability, as energy harvesting is subjected to uncontrollable environmental factors. This allows neighbouring nodes that use it as their next-hop relay node to synchronize their duty cycles with it as best as their own energy harvesting characteristics or profiles permit. The duty cycles are thus determined by energy harvesting profiles of the sensors and the amount of traffic carried by each node is also implicitly determined by its energy harvesting profile, i.e. a node that wakes up more often because it is able to harverst more energy will end up transmitting (relaying) more data packets. The use of energy harvesting introduces uncertainty in the energy availability which is not addressed by receiver-initiated asynchronous duty cycling techniques for battery-powered WSNs [6][7]. To deal with this uncertainty, we adopt fuzzy control techniques to determine optimal duty cycles for nodes. This achieves load and energy balancing across the nodes depending on their energy harvesting characteristics, thus, its name – Load and Energy Balancing (LEB) MAC protocol, or LEB-MAC.

This paper is organized as follows. Section 2 presents and compares the related work on MAC protocols for EH-WSNs. The details of the proposed LEB-MAC are given in Section 3, followed by the performance evaluation of LEB-MAC in Section 4, based on the performance metrics like end-to-end delay, sender duty cycle, packet delivery ratio, collision rate, and fairness index. We also evaluate how quickly it can replenish its energy supply and continue operating under realistic conditions. Finally, Section 5 concludes this paper.

## II. Related Work

The focus of our research is duty cycling mechanisms in MAC protocols for wireless sensor networks that are powered by energy harvesting. Recent advances in energy harvesting technology have motivated the emergence of energy harvesting powered wireless sensor networks or EH-WSNs. Solar powered WSNs have been the main focus of research in EH-WSNs as photovoltaic EH technology is one of the most matured and able to meet performance expectations at WSN form factors [2][8]. Other sources of ambient energy are also being actively studied [1][2], e.g. vibration, wind, heat, and most recently, radio waves [9]. Here, we review prominent WSN duty cycling schemes that show potential for use in EH-WSNs as well as the recent related work on EH-WSNs.

Duty cycling, which efficiently saves energy by removing unnecessary idle listening and overhearing, has been a primary mechanism in MAC schemes for battery-powered WSNs. However, the efficiency of duty cycle based WSNs relies heavily on awareness of the neighbourhood nodes' schedules. With the knowledge of neighbours' schedules, a sensor is therefore able to avoid collisions. To achieve this awareness goal, Cao *et al*. [10] propose a staged and smooth transition procedure for updating schedules to ensure that all one-hop neighbours are able to successfully receive updated schedules. Through accurate knowledge of neighbours' up-to-date schedules, the staged and smooth transition approach successfully avoids the logical isolation problem and efficiently achieves a trade-off between energy cost and system performance.

Duty cycle based MAC protocols are classified into two categories: synchronous and asynchronous. Synchronous duty cycle based MAC protocols, such as S-MAC [11], T-MAC [12], DW-MAC [13], and SCP [14], synchronize the sleep and duty schedules among sensors to avoid idle listening, overhearing and collisions. S-MAC is one of the earlier synchronous duty cycle MAC protocols proposed for WSNs. S-MAC divides time into cycles and each cycle is further divided into three periods: Sync, Data, and Sleep. The Sync period is for clock synchronization among nodes. Nodes transmit data packets during Data periods and power off during Sleep periods. Before transmitting data packets, request-to-send and clear-to-send packets are exchanged first, followed by the data packet and finally the Acknowledgement frame. A multi-hop route requires multiple cycles to deliver a packet and consequently incur long end-to-end delay. T-MAC has been designed to handle variable loads by dynamically shortening the Data period when there is no data transmission. T-MAC shows comparable performance with S-MAC under homogeneous load conditions, but when tested under variable load scenarios, it outperforms S-MAC by a factor of 5. WSNs can be subjected to bursty and high traffic loads for certain periods, e.g., data broadcast. Instead of transmitting data packets during the Data period, DW-MAC broadcasts a scheduling frame in the Data period to establish the times for transmitting data packets during the Sleep period that follows. DW-MAC shortens Data periods and lengthens Sleep periods, for all nodes except those involved in the packet exchange; this reduces latency and energy usage. Scheduled Channel Polling (SCP) eliminates the need for long preambles like in low power listening (LPL) by synchronizing nodes with periodic broadcast of synchronization packets, like in S-MAC. The cost of periodic synchronization, however, reaps the benefit of requiring only a very short wakeup tone to wake up receivers and significant energy savings. As packets need to be transmitted across multiple hops to reach the sink, WSN nodes would have to maintain multiple schedules, thus increasing the complexity. Fixed transmission schedules also do not suit the statistical nature of traffic and are extremely hard to maintain when energy availability is intermittent and unpredictable, like in EH-WSNs.

Asynchronous duty cycle based MAC protocols utilize pre-transmission cordination, i.e., before transmitting packets, senders and receivers first communicate with one another to synchronize. The pre-transmission coordination may be either *sender-initiated*, by sending a preamble to notify the receiver of oncoming of packets, or *receiver-initiated*, where a receiver sends a wakeup beacon to solicit for packets from receivers. In B-MAC [15] and X-MAC [16], which are among the earlier MAC protocols with sender-initiated asynchronous duty cycle mechanisms, each node periodically wakes up to check whether the channel is active or not. If the channel is active, the node stays awake to receive packets that may be destined to it. In order to coincide with receivers' active cycle, a sender with pending data packets first transmits a preamble of duration longer than the receiver's periodic wakeup interval. X-MAC uses the data frame itself as the preamble, which simplifies the process and helps a sender determine whether the data frame is successfully received or not. WiseMAC [17] is the pioneer in predictive wakeup using fixed wakeup intervals. Since each node wakes up in fixed intervals, a sender can easily to predict a receiver's wakeup time in the next duty cycle and shorten the required preamble length. However, fixed wakeup intervals can result in collisions of simultaneous transmission of preambles.

Lastly, we have the receiver-initiated asynchronous MAC protocols. In RI-MAC [6], each node periodically wakes up and broadcasts a beacon to notify neighbours of its active state and readiness to receive packets. A node with data packets to transmit wakes up to wait for the beacon from the receiver. Once it hears the beacon, it sends the data packet and waits for the receiver's acknowledgment of successful receipt. Since a wakeup beacon is substantially shorter than a preamble, the receiver-initiated protocols typically outperform the sender-initiated ones. In comparison to RI-MAC longer senders' duty cycles arising from the need for senders to wake up to listen as as soon as there are packets to send, PW-MAC [7] senders rely on a predictive wakeup mechanism to wakes up right before the intended receiver wakes up. It utilizes pseudo random wakeup schedules instead of fixed schedules to avoid the collision probability. Furthermore, the use of a pseudo random number generator enables each node to compute and precisely predict the wakeup schedule of a receiver. Thus, PW-MAC shortens sender duty cycles and reduces collisions.

Unlike synchronous MAC protocols, no dissemination of wakeup intervals for asynchronous MAC protocols is required because a sender/receiver is able to predict the wakeup time by either fixed schedule or pseudo random generator. However, asynchronous MAC protocols still require sensors to have knowledge of wakeup intervals among neighbouring sensors in order to send preambles and/or beacons. Existing asynchronous duty cycle MAC protocols utilize either fixed or random wakeup intervals. Recently, Yoo *et al*. [18] propose a dynamic duty cycle scheduling scheme, called Duty-cycle Scheduling based on Residual energy (DSR), that adjusts the duty cycles based on nodes' residual energy. This effectively reduces the average end-to-end latency, improves the packet delivery ratio, and prolongs the lifetime of sensor networks; however, they do not consider the load balancing.

This paper presents a new asynchronous receiver-initiated MAC protocol with energy and load balancing for EH-WSN. Due to the feature of energy and load balancing, the proposed LEB-MAC outperforms in many aspects, namely, low receiver and sender duty cycle, high throughput, high fairness, and low end-to-end delay.

## III. The LEB-MAC Protocol

This section presents the design of the LEB-MAC protocol for EH-WSNs which comprises of the following four parts: receiver-initiated duty-cycling protocol, energy-aware interval design, automatic load-Energy balancing and collision resolution schemes.

### *A. Receiver-initiated transmission*

A receiver-initiated duty-cycling protocol not only shortens the duty cycles of the receiver and sender but also reduces the probability of collision. In addition, a duty-cycling protocol, which interleaves active and sleep periods, provides a good opportunity for energy harvesters to recharge the energy storage before going into a discharging state where the stored energy is consumed by the sensor node. This is the motivation for the LEB-MAC protocol to adopt a receiver-initiated duty-cycling protocol.

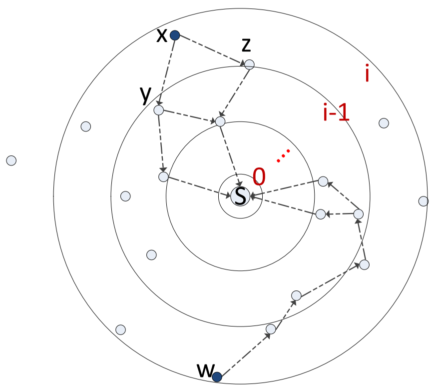



Fig. 1. An illustrated example of a wireless sensor network

Consider a WSN as shown in Fig. 1. We rank the sink as level 0, and then rank the sensor nodes' level based on the number of hops needed to reach the sink. That is, one-hop neighbours of the sink ranked as level 1, nodes that are two-hops away from the sink are ranked as level 2, and so on. As packets are sent toward the sink, without loss of generality, if there is no route detour, packets are sent by a node of level *i* to its upstream neighbour(s) of level *i*-1. For a node *x* of level *i*, the nodes of level 0<*j*<*i* located in the path from the node *x* to the sink are the upstream nodes of *x*. Thus, as shown in Fig. 1, a receiver node (*y* or *z*) of level *i*-1 is an immediate upstream neighbour of the sender node *x* of level *i*. An example of a route detour (moving away from the sink) in Fig. 1 is the path from node *w* of level *i* to the sink. A node will always attempt to send its packet to an upstream receiver first, failing which, it will send it to a receiver at the same level, and lastly, to a downstream receiver if both previous options failed.

For receiver-initiated duty-cycling protocol, a receiver first sends a beacon to invite its downstream neighbouring nodes to send data packets, which is akin to polling. To successfully receive a beacon so as to synchronize with the receiver and send data packets, a sender has to wake up before a beacon's transmission. Thus, a sender needs to know its upstream neighbours' wakeup schedules, which can be either notified by previous beacons, or predicted using pseudo random or fixed wakeup schedules. A beacon mainly notifies senders that the receiver is ready to receive data packets. Sometimes, it also conveys the wakeup time of next duty cycle. Upon hearing a beacon, a sender responds by transmitting a data packet if it has packets to send. After the receiver successfully receives the data packet, it sends an acknowledgement (ACK) to inform the sender that it has successfully received the packet before going back to sleep again. If multiple downstream neighbours respond to the beacon with data packets simultaneously, the packets collide at the receiver which receives a grabled signal and it responds with a negative acknowledgement (NACK). We denote such a collision as a *data collision*. When the senders receive a NACK, they are aware of the failed transmission and invoke the data-collision resolution procedure (described in the Subsection D). Fig. 2 shows the basic operation of receiver-initiated duty-cycling MAC protocol.

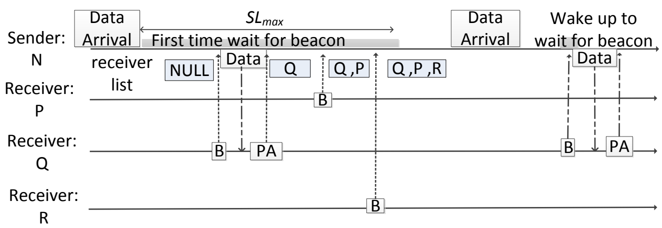



Fig. 2. Basic operation of receiver-initiated duty-cycling protocol

In LEB-MAC, the beacon also conveys the wakeup interval of next duty cycle. For a sender that is listening for a particular receiver's beacon for the first time, it does not have any prior information to tell it when to wake up to wait. Therefore, it has to stay awake long enough, possibly as long as a maximum listening period ($SL_{max}$) to listen for beacons and acquire the needed information of the upstream receiver's schedules; $SL_{max}$, defined in the next subsection, is the maximum sleep duration of a (receiver) node with low energy. We note that a node wishing to send a packet can only remain awake as long as permitted by its available energy, and may still miss a beacon. In the Fig. 2 example, the receiver list of node N is null if node N is communicating for the first time after a *cold start*; an EH-WSN node begins from a *cold start* when it has previously expended all its stored energy and lost all state information. It is important to note that even when an EH-WSN node is not operating, it still loses its stored energy through leakage, and if it is unable to harvest energy faster than the rate of leakage, it will inevitably lose all its energy and state information. After receiving beacons from receivers Q, P and R, the receiver list of node N becomes (Q, P, R). With the acquired information on upstream nodes' schedules, node N now knows when to wake up to wait for the next beacon. For subsequent packet transmissions, node N wakes up just before the expected beacon transmission time of a receiver, as shown in Fig. 2. However, though rare, there might be multiple receivers sending beacons simultaneously, resulting in what we denote as *beacon collision*. A beacon collision resolution (as well as, *missed beacon*) procedure is therefore invoked to resolve this, details of which are described in Subsection D.

The LEB-MAC frame format is adapted from that of IEEE802.15.4 as shown in Fig. 3. The frame control field describes the frame type: beacon, acknowledgement or data frame as shown in TABLE I.

TABLE I. FRAME CONTROL FIELD VALUES

| Frame control | Description |
|---|---|
| 000 | Beacon |
| 001 | Data |
| 010 | Positive ACK – ACK |
| 011 | Negative ACK – NACK |
| Others | Reserved |

For the frame format of beacon, to convey the wakeup time of next cycle, a 4-octet "Interval" field is inserted as shown in Fig. 3(a). In the data frame, a load field with four octets is augmented to notify the receiver of the sender's load level as shown in Fig. 3(b). The receiver then uses this information to allocate appropriate time slots for different senders.

| Octets: 1 | 1 | 4 | 4 | 4 | 2 |
|---|---|---|---|---|---|
| Sequence number | Frame control | Source Address | Destination Address | Interval | FCS |
| MHR(MAC Header) | | | | MAC payload | MFR |

(a) format of Beacon and ACK frame

| Octets: 1 | 1 | 4 | 4 | 4 | variable | 2 |
|---|---|---|---|---|---|---|
| Sequence number | Frame control | Source Address | Destination Address | Load | Data payload | FCS |
| MHR(MAC Header) | | | | MAC payload | | MFR |

(b) format of Data frame

Fig. 3. LEB-MAC frame formats

### *B. Energy-aware duty cycle*

LEB-MAC determines the active-sleep cycle according to the energy level of receiver nodes, which depends on rate at which energy can be harvested; the energy harvesting rate is dependent on both the harvester characteristics as well as environmental conditions. A node dynamically plans its own wakeup schedule according to its own current energy level. Nodes with higher energy supplies wake up more frequently, with shorter sleep intervals, and therefore receive and relay more packets. Receiving more data packets means bearing more of the overall network traffic load. This feature naturally makes the LEB-MAC protocol distribute loads according to nodes' energy. Thus, such an energy-aware duty cycle not only reduces the number of nodes facing energy shortage but also provides load-balancing among relay nodes. Obviously, in order for this feature to be effective, there must be enough nodes deployed to provide multiple upstream relays. Otherwise, if a particular node is the only upsteam relay for a group of downstream nodes, then all traffic will have to be sent through it. In addition, to avoid collisions arising from neighbouring nodes waking up at the same time and contending for the channel, LEB-MAC consciously selects sleep durations based on nodes' energy availability which can never be identical for any arbitrary pair or group of nodes.As aforementioned, a node with lower energy level chooses a longer sleep interval to harvest as much energy as possible in order to maintain at a certain energy level. For determining the next sleep interval, we adopt the concept of a fuzzy logic controller [19] and define a fuzzy term set for node energy, $T(e)$ = {Low($L$), Medium($M$), High($H$)}. More energy levels can be defined to provide finer granularity of control if required by an application. In order to determine appropriate sleep intervals according to node energy level, we define the term set of output controls as $T(y)$= {Sleep Low($SL$), Sleep Medium($SM$), Sleep High($SH$)}, where $SL$, $SM$, and $SH$ are the corresponding output control actions for low, medium and high energy levels. The membership functions for terms in the set are specified by a triangular function $f(x;x_0,a_0,a_1)$ or a trapezoidal function $g(x;x_0,x_1,a_0,a_1)$ as these functions are suitable for real-time operation. They are generally defined by the following equations:

$$f(x;x_0,a_0,a_1)=\begin{cases}\dfrac{x-x_0}{a_0}+1 & \text{for } x_0-a_0<x\le x_0\\ \dfrac{x_0-x}{a_1}+1 & \text{for } x_0<x\le x_0+a_1\\ 0 & \text{otherwise}\end{cases} \quad (1)$$

$$g(x;x_0,x_1,a_0,a_1)=\begin{cases}\dfrac{x-x_0}{a_0}+1 & \text{for } x_0-a_0<x\le x_0\\ 1 & \text{for } x_0<x\le x_1\\ \dfrac{x_1-x}{a_1}+1 & \text{for } x_1<x\le x_1+a_1\\ 1 & \text{otherwise}\end{cases} \quad (2)$$

where $x_0$ in $f(\bullet)$ is the center of the triangular function; $x_0(x_1)$ in $g(\bullet)$ is the left (right) edge of the trapezoidal function; $a_0(a_1)$ is the left (right) width of the triangular or the trapezoidal function. Let $\mu_L(e)$, $\mu_M(e)$ and $\mu_H(e)$ denote the membership functions of $L$, $M$ and $H$ in $T(e)$, respectively. They are defined by the following three equations:

$$\mu_L(e)=g(e;0,L_e,0,L_w) \quad (3)$$

$$\mu_M(e)=f(e;M_c,M_{w0},M_{w1}) \quad (4)$$

$$\mu_H(e)=g(e;H_e,1,H_w,0) \quad (5)$$

As shown in Fig. 4, the values of $L_e$, $M_c$ and $H_e$ are usually set as the low, medium and high thresholds, respectively. The intervals $L_w = M_{w0} = M_c - L_e$ and $H_w = M_{w1} = H_e - M_c$ could be the difference between the two thresholds. The membership functions associated with the terms $SL$, $SM$, and $SH$ in $T(y)$ are denoted by $\mu_{SL}(y),\mu_{SM}(y)$ and $\mu_{SH}(y)$, respectively, which are given by the following equations:

$$\mu_{SL} = g(y; 0, SL_c, 0, SL_w) \times SL_d + SL_{min} \tag{6}$$

$$\mu_{SM} = g(y; 0, SM_c, 0, SM_w) \times SM_d + SM_{min} \tag{7}$$

$$\mu_{SH} = g(y; 0, SH_c, 0, SH_w) \times SH_d + SH_{min} \tag{8}$$

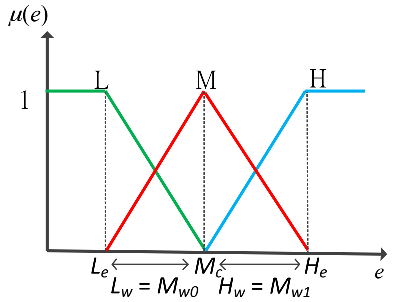



Fig. 4. The definitions of functions $\mu_L(e), \mu_M(e)$ and $\mu_H(e)$ (cf: [19] Fig. 1.)

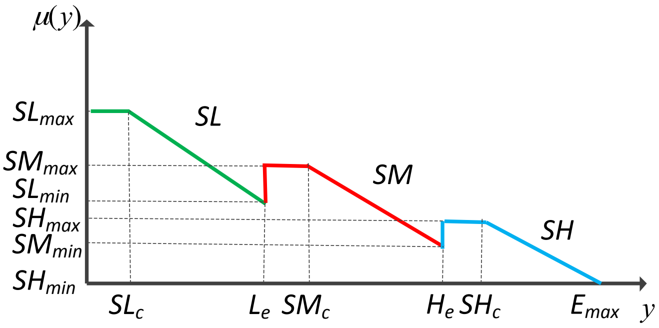



Fig. 5. The definitions of functions $\mu_{SL}(y), \mu_{SM}(y)$ and $\mu_{SH}(y)$ .

We extend the single level interval setting used by DSR [18] to determine the sleep interval for LEB-MAC's three levels. With reference to Fig. 5, we make the following definitions. For energy levels $L$, $M$ and $H$, maximum and minimum limits for the sleep intervals $SL_{max}$, $SL_{min}$, $SM_{max}$, $SM_{min}$, $SH_{max}$, and $SH_{min}$, are given, respectively, where $SL_{max} > SM_{max} > SH_{max}$. In order to meet the minimum quality of service requirement of applications, $SL_{max}$ is an application-specific parameter. We let $SL_d = SL_{max} - SL_{min}$, $SM_d = SM_{max} - SM_{min}$ and $SH_d = SH_{max} - SH_{min}$. For each energy level $L$, $M$ or $H$, the sleep interval is set to the maximum limit as the energy level is less than or equal to the predetermine threshold $SL_c$, $SM_c$, and $SH_c$, respectively. The $SL_w$, $= L_e - SL_c$, $SM_w = H_e - SM_c$, and $SH_w = E_{max} - SH_c$, respectively, where $E_{max}$ represents the maximum energy that can be stored on a node. As shown in Fig. 5, the $SH_{min}$ is set to 0 and therefore $SH_d = SH_{max}$.

## C. Automatic Load-Energy balancing

An arbitrary sender may have multiple upstream nodes and therefore multiple routes toward the sink. LEB-MAC performs automatic load distribution among all the upstream neighbours according to their energy levels since receivers wake up according to their energy levels, and a receiver with higher energy wakes up more times and consequently relays more packets. Nodes with packets to send will transmit them to one of their upstream neighbours that are active to receive packets. No balancing control mechanism is required for sending nodes to achieve the load-energy balancing. Contention among multiple senders will be resolved by collision resolution mechanisms, to be discussed next.

## D. Collision resolution

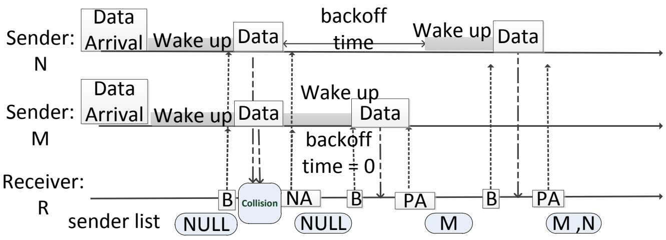
As aforementioned, there are two types of collisions: data collision and beacon collision. The *data collision* is caused by multiple senders responding to a receiver while the *beacon collision* is caused by multiple receivers simultaneously issuing beacons to invite senders to transmit their packets. We present different collision resolution procedures to address these two conditions.

- Data collision: These are further classified into two cases: (i) all senders transmitting for the first time and (ii) some senders have previously communicated with the receiver.

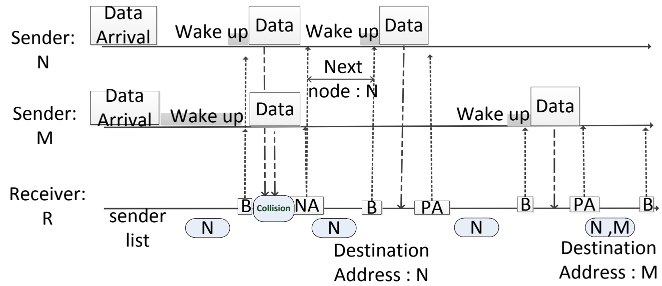
  Case (i) If all senders are communicating with the receiver for the first time and collision occurs, the receiver replies with a NACK frame to notify the senders that their transmissions have collided, as shown in Fig. 6(a). When the senders receive the NACK, each of them selects a random backoff time depending on their energy level, starts the backoff timer and goes back to sleep. After the backoff timer expires, the sender wakes up to wait for a beacon and repeat the procedure as shown in Fig. 2 to send its data packet.

  Case (ii) Similar to senders maintaining a receiver list, a receiver also maintains a "sender list" of nodes which have been communicating with it in the previous transmission cycles. If there is at least one node already communicating with the receiver, such as the sender N in Fig. 6(b), it will be in the sender list and in the following cycle, a dedicated beacon for node N (cf: Fig. 3(a), Destination Address = N; else, Destination Address = 0xFFFFFFFF for broadcast) is sent to reserve next cycle for node N. Since a dedicated beacon to node N is also heard by the sender M, it can either try to send the packet to other receivers or wait for the next duty cycle after the cycle for the node N. When there are multiple nodes in the sender list, a receiver will utilize the information obtained from the Load field of the previous Data Frames (Fig. 3(b)) to schedule the various senders over the next few duty cycles. Senders can indicate, e.g. the priority of their data, whether they have more segments to send like the More Fragments bit in TCP headers, etc. They can also indicate that their current stream of packets have ended so that the receiver knows and will give other senders the transmission opportunity in the next cycle.

- Lost/Missed Beacon or Beacon collision: The same resolution procedure is invoked to deal with a *missed beacon* or *beacon collision*. Due to lack of prior duty cycle knowledge or clock drift that put a sender node's clock out of sync with its receivers, it may miss a beacon from a receiver. A sender may also receive a malformed signal due to beacon collision or transmission error. In any case, the sender remains awake to wait for the next valid beacon from the receiver(s). When a receiver transmits a beacon and does not get a respond, it goes back to sleep assuming that there is no node wanting to transmit. Since the sleep intervals are determined by a receiver's energy level and the probability of two neighbouring receivers having synchronized sleep intervals is extremely low, there is high probability that two receivers send beacons at different times in the next cycle. Furthermore, a receiver that did not get any response to its beacon would not have expended much energy and likely to sleep for a short period before coming awake again to solicit for packets with a beacon. The waiting sender can then respond to the new beacon and synchronize with the receiver.




(a) Illustration of data collision with all nodes communicating the first time.




(b) Illustrationof data collision with some nodes already communicating.

Fig. 6. Data collision examples

## IV. Performance Evaluation

To validate the proposed LEB-MAC and evaluate its performance, we conducted simulation studies using Qualnet. The simulated network is a 5×5 grid with 80% node density (Fig. 7 shows various configurations with node transmission range which is the same for all nodes.) Key parameter values are listed in TABLE II. , where $\lambda_{flow}$ is per flow arrival rate.

TABLE II. KEY PARAMETER VALUES

| | | | |
|---|---|---|---|
| $L_e$ | 30% | $SL_c$ | 10% |
| $M_c$ | 50% | $SM_c$ | 35% |
| $H_e$ | 70% | $SH_c$ | 75% |
| $E_{max}$ | 100% | $\lambda_{flow}$ | 1 pkt/s |
| $SL_{max}$ | 2.0s | $SL_{min}$ | 1.0s |
| $SM_{max}$ | 1.5s | $SM_{min}$ | 0.5s |
| $SH_{max}$ | 1.0s | $SH_{min}$ | 0s |
| Note: % – percentage of a node's storage capacity | | | |

We compare LEB-MAC against RI-MAC [6], PW-MAC [7], and RI-MAC with DSR [18] (denoted as RI-DSR), all operating under solar energy harvesting. First, we studied the protocols using same network settings as [18] We use Cymbet's solar energy harvester CBC-EVAL-09 [20] as the energy source and an initial energy level for all sensors to be 25%, the results of 180-sec simulation runs are presented in Fig. 8. Next, we used the 24-hr empirical data for solar energy harvesting presented in [21]. Results for sunny and cloudy days are shown in Fig. 9 and Fig. 10, respectively.

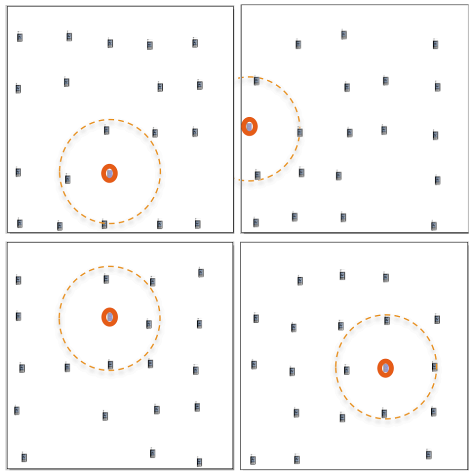
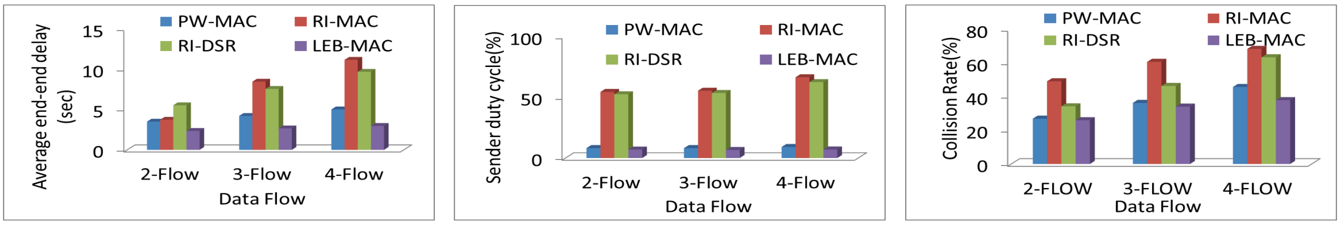


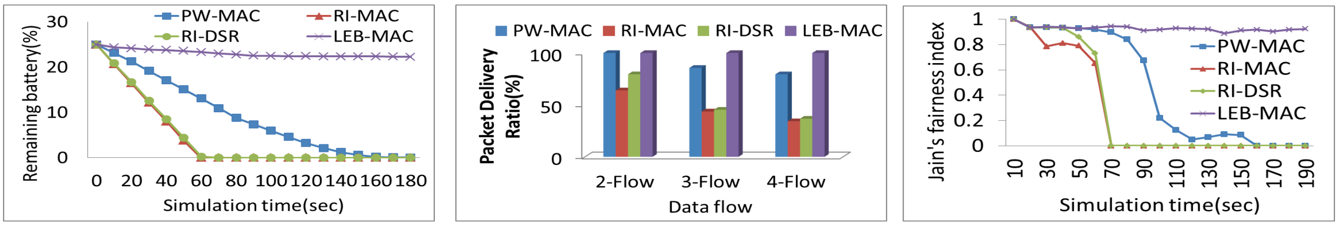
Fig. 7. Examples of network topologies for simulation study

The results shown in Fig. 8 are for a scenario aimed at evaluating the functionality of LEB-MAC. Since the initial energy level is set at 25%, there may be fewer nodes active, more network holes and route detours. While LEB-MAC shows the best overall performance, we note that PW-MAC's predictive wakeup mechanism was able keep the senders' duty cycles short, saving substantial energy. Longer sender duty cycles (Fig. 8(b)) of RI-MAC and RI-DSR manifested in higher energy usage, more network holes as nodes need longer sleep cycles, and consequently, higher end-to-end delays (Fig. 8(a)). Both these protocols quickly depleted their energy storage (Fig. 8(d)) and could not recover thereafter. Starting with 25% energy level puts many nodes in longer sleep cycles and leaves the network with fewer routes, and this increases the contention among flows (Fig. 8(c)). While both LEB-MAC and PW-MAC are able to achieve higher packet delivery ratio (PDR) than the other protocols, LEB-MAC's performs better and maintains 100% PDR under heavier load conditions (3- and 4-flows) as shown in Fig. 8(e). Fig. 8(f) shows the fairness index [22] of the protocols, which measures how well they are able to allocate resources equally among all network nodes and deliver their packets. Through effective energy and load balancing, LEB-MAC is able to sustain a fairness index close to 1. In theory, EH-WSN nodes should be able to replenish their energy supply over time. However, in this particular scenario, the energy harvesting rate was too low for the other protoocls to effectively replenish; only LEB-MAC continued operating with energy level around 25%.

Next, we used realistic empirical EH data measured over 24hrs [21], starting at midnight, with nodes having an initial energy level of 75%. As expected, the nodes were unable to harvest any energy during darkness and exhausted their energy supply quickly. However, by 07:30hrs (~27000s) on a sunny day, the nodes quickly replenished their energy supply (Fig. 9(d)) and continued harvesting enough energy to operate until sunset around 18:00hrs (64800s). With adequate sunlight, the nodes were able to stay active longer, providing more routes to the sink. This reduced the contention (lower collision rate, as shown in Fig. 9(c)) and improved performance in general (Fig. 9(a), (b) and (f).) LEB-MAC was able to retain energy longer (by 1.5hrs) before energy levels are depleted. PDR is less than 100% because it is averaged over the entire 24hr period.

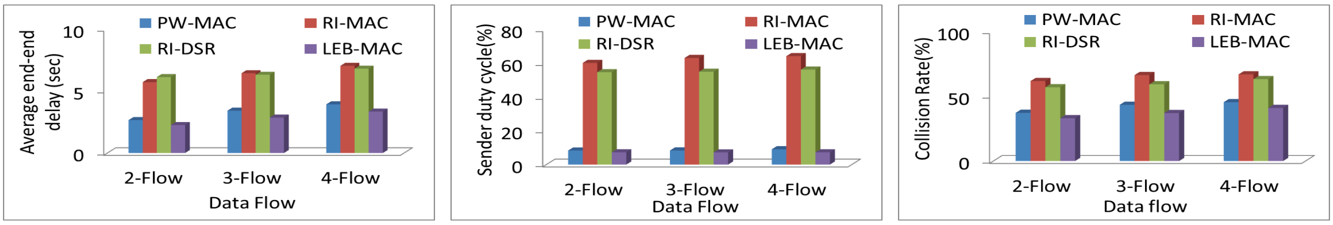



(a) average end-to-end delay vs. data flows (b) sender duty cycle vs. data flows (c) collision ratio vs. data flows

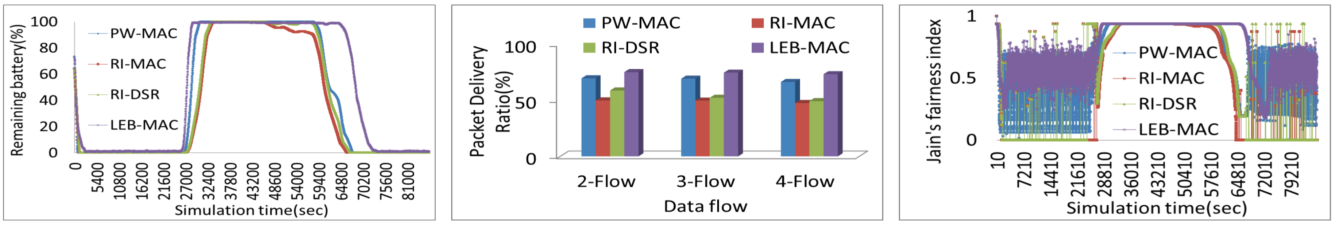



(d) average remaining energy vs. data flows (e) packet delivery ratio vs. data flows (f) fairness index vs. simulation time

Fig. 8. RI-MAC, RI-DSR, PW-MAC and LEB-MAC performance, with 25% initial energy level.

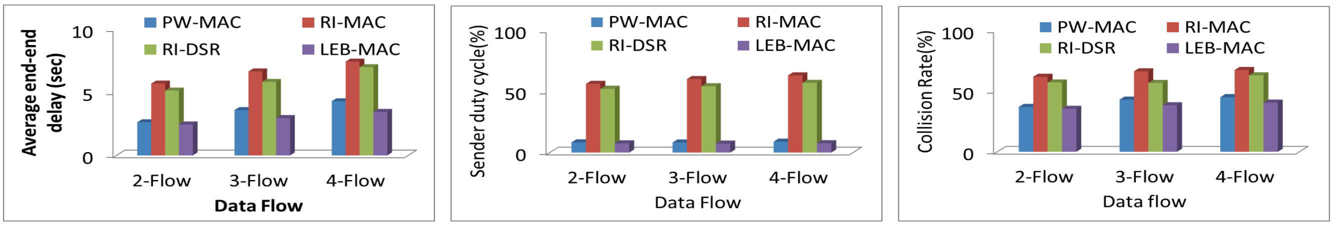



(a) average end-to-end delay vs. data flows (b) sender duty cycle vs. data flows (c) collision ratio vs. data flows

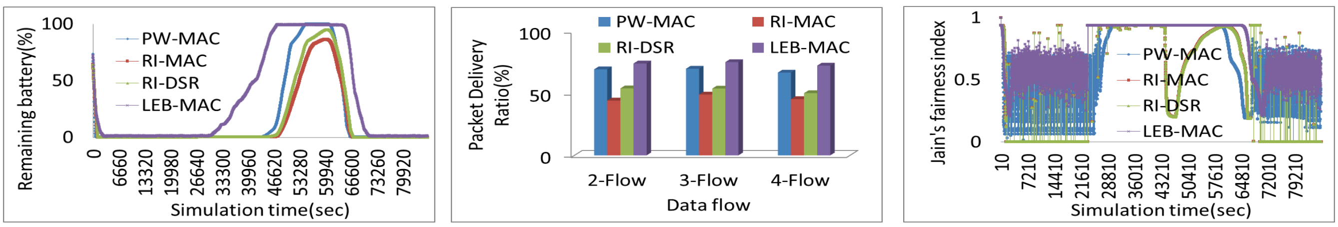



(d) average remaining energy vs. data flows (e) packet delivery ratio vs. data flows (f) fairness index vs. simulation time

Fig. 9. RI-MAC, RI-DSR, PW-MAC and LEB-MAC performance under sunny day conditions




(a) average end-to-end delay vs. data flows (b) sender duty cycle vs. data flows (c) collision ratio vs. data flows




(d) average remaining energy vs. data flows (e) packet delivery ratio vs. data flows (f) fairness index vs. simulation time

Fig. 10. RI-MAC, RI-DSR, PW-MAC and LEB-MAC performance under cloudy day conditions

Lastly, we simulated the protocols using empirical data for energy harvesting performance on a cloudy day with rain in the morning [21]. We observe from the energy levels (Fig. 10(d)) that it took longer time since sunrise for the nodes to replenish their energy supply, with LEB-MAC being the fastest to reach 100%, PW-MAC needing another 1hr to 1.5hrs, while RI-MAC and RI-DSR never reaching full energy levels because of their energy intensive longer sender duty cycles (Fig. 10(b)) and higher collision rates (Fig. 10(c)).

Under all the scenarios studied, LEB-MAC and PW-MAC are able to achieve much better performance (subfigures (a) and (e) of Fig. 8, Fig. 9 and Fig. 10) with shorter sender duty cycles (Fig. 8(b), Fig. 9(b) and Fig. 10(b)). RI-DSR's lack of load balancing is highly evident in the fairness index (Fig. 9(f) and Fig. 10(f)) where it exhibited poor performance except in very ideal conditions. While all protocols are able to operate under good light conditions, only LEB-MAC is able to continue operating under poor energy harvesting conditions.

## V. Conclusion

This paper has proposed a load and energy balancing receiver-initiated duty cycle MAC protocol (LEB-MAC) for energy harvesting powered wireless sensor networks. The simulation results reveal that LEB-MAC outperforms in many aspects, including end-to-end delay, sender duty cycle, collision ratio, packet delivery ratio and fairness index. The inherent load-balancing feature also enables LEB-MAC to continue operating in challenged conditions (poor energy harvesting) where other protocols failed. It is envisaged that in very dense EH-WSNs with highly correlated energy harvesting patterns (e.g. in a railroad monitoring WSN [23], vibration-powered nodes in close proximity to one another all charging simultaneously and acquiring sensor data that needs to be sent immediately) mechanisms like probabilistic polling [3] may be needed to alleviate contention among nodes.

## References


[1] W. K. G. Seah, Z. A. Eu and H. P. Tan, "Wireless Sensor Networks Powered by Ambient Energy Harvesting (WSN-HEAP) – Survey and Challenges," in *Proceedings of the 1st International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology* (Wireless VITAE), Aalborg, Denmark, 17-20 May 2009, pp. 1–5.

[2] M. A. Ingram *et al*., "Perspectives on Energy-Harvesting Wireless Sensor Networks," in *Globalization of Mobile and Wireless Communications: Today and in 2020*, ser. Signals and Communication Technology, R. Prasad *et al*., Eds. Springer, 2011, ch. 17, pp. 249–274.

[3] Z. A. Eu, W. K. G. Seah and H. P. Tan, "A study of MAC schemes for wireless sensor networks powered by ambient energy harvesting," in *Proceedings of the 4th Annual International Conference on Wireless Internet* (WICON), Maui, Hawaii, USA, 17-19 Nov 2008, pp. 78:1–9.

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

[5] R. Shigeta, Y. Kawahara and T. Asami, "Capacitor Leakage Aware Duty Cycle Control for Energy Harvesting Wireless Sensor Networks," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, Seattle, WA, USA, 1-4 Nov 2011, pp. 387–388.

[6] Y. Sun *et al.*, "RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, Raleigh, NC, USA, 4-7 Nov 2008, pp. 1-14.

[7] T. Lei *et al.*, "PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks," in *Proceedings of IEEE INFOCOM*, Shanghai, China, 10-15 April 2011, pp. 1305-1313.

[8] K. Lin *et al.*, "Heliomote: enabling long-lived sensor networks through solar energy harvesting," in *Proceedings of the 3rd international conference on Embedded networked sensor systems*, San Diego, California, USA, 2005, pp. 309-309.

[9] A. P. Sample, B. H. Waters, S. T. Wisdom and J. R. Smith, "Enabling Seamless Wireless Power Delivery in Dynamic Environments," *Proceedings of the IEEE*, Vol 101, No 6, pp. 1343-1358, June 2013.

[10] Y. Cao, Z. Zhong, Y. Gu and T. He, "Safeguarding schedule updates in wireless sensor networks," in *Proceedings of IEEE INFOCOM Mini-Conference*, 2011, pp. 606-610.

[11] Y. Wei *et al.*, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of IEEE INFOCOM*, New York, USA, 23-27 June 2002, pp. 1567-1576.

[12] T. v. Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, Los Angeles, California, USA, 2003, pp. 171-180.

[13] Y. Sun*, et al.*, "DW-MAC: a low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks," in *Proceedings of the 9th ACM International Symposium on Mobile ad hoc networking and computing*, Hong Kong, China, 2008, pp. 53-62.

[14] W. Ye*, et al.*, "Ultra-low duty cycle MAC with scheduled channel polling," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, Boulder, Colorado, USA, 2006, pp. 321-334.

[15] J. Polastre*, et al.*, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded networked sensor systems*, Baltimore, MD, USA, 2004, pp. 95-107.

[16] M. Buettner, *et al*., "X-MAC: a short preamble MAC protocol for duty cycled wireless sensor networks," in *Proceedings of the 4th international conference on Embedded networked sensor systems*, Boulder, Colorado, USA, 2006, pp. 307-320.

[17] A. El-Hoiydi and J.-D. Decotignie, "WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks," in *Algorithmic Aspects of Wireless Sensor Networks*, 2004, pp. 18-31

[18] H. Yoo, M. Shim and D. Kim, "Dynamic Duty Cycle Scheduling Schemes for Energy-Harvesting Wireless Sensor Networks," in *IEEE Communications Letters*, 2012, pp. 202-204.

[19] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller. I," in *IEEE Transactions on Systems, Man and Cybernetics,* Vol 20, No 2, Mar/Apr 1990, pp. 404-418.

[20] Cymbet, "EnerChip EP Universal Energy Harvester Eval Kit," CBC-EVAL-09, http://www.cymbet.com/pdfs/DS-72-13.pdf.

[21] C. K. Ho, P. D. Khoa and C. M. Pang, "Markovian models for harvested energy in wireless communications," in *Proceedings of the IEEE International Conference on Communication Systems* (ICCS), Singapore, 17-19 Nov 2010, pp.311-315.

[22] R. Jain, D. M. Chiu and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems," *DEC Research Report TR-301*, 1984.

[23] H. P. Tan, P. W. Q. Lee, W. K. G. Seah and Z. A. Eu, "Impact of Power Control in Wireless Sensor Networks Powered by Ambient Energy Harvesting (WSN-HEAP) for Railroad Health Monitoring," in *Proceedings of the International Conference on Advanced Information Networking and Applications Workshops*, Bradford, UK, 26-29 May 2009, pp. 804-809.