

# LB-MAC: A Lifetime-Balanced MAC Protocol for Sensor Networks

Yang Peng, Zi Li, Wensheng Zhang, and Daji Qiao

Iowa State University, Ames, IA 50011, USA  
{yangpeng,zili,wzhang,daji}@iastate.edu

**Abstract.** This paper presents LB-MAC, a new MAC protocol for asynchronous, duty cycle sensor networks. Different from existing sensor network MAC protocols that usually focus on reducing energy consumption and extending lifetime of individual sensor nodes, LB-MAC aims at prolonging the network lifetime through balancing the nodal lifetime between neighboring sensors. LB-MAC is lightweight and scalable as the required control information is only exchanged locally between neighbors. LB-MAC has been implemented in TinyOS and evaluated on a sensor network testbed with extensive experiments. Results show that LB-MAC is able to achieve a significantly longer network lifetime than state-of-the-art MAC protocols such as X-MAC, RI-MAC and SEESAW, while maintaining comparable levels of network power consumption, packet delay and delivery ratio.

## 1 Introduction

Energy conservation is perhaps the most important issue in battery-operated sensor networks. It is always desirable to extend the operational lifetime of a sensor network as much as possible. For many sensor network applications [1,2], the network lifetime is often defined as the minimal nodal lifetime among all sensor nodes in the network. This is because, the depletion of battery energy of bottleneck sensor nodes, such as the nodes close to the root node in a tree topology network, may cause network disconnection, which could render the sensor network nonfunctional. Although energy saving techniques such as energy-aware routing can be used to reduce the workload and extend the lifetime of bottleneck sensor nodes, they may still consume higher energy than other nodes in the network and thus bound the network lifetime. Besides, sensor nodes with a similar level of workload may have different nodal lifetime due to environmental or system reasons. For example, nodes with poorer-quality batteries or solar-rechargeable nodes deployed to shady locales may have shorter lifetime than their peers. Therefore, to maximize the network lifetime, it is important to extend the lifetime of individual sensor nodes or, to be more specific, the shortest nodal lifetime among all nodes.

Despite the need for a holistic approach to the energy conservation issue and to prolonging the network lifetime, most of the current MAC protocol design has been focusing on reducing the energy consumption and extending the operational

lifetime of individual sensor nodes. To remedy this deficiency, we investigate the MAC protocol design from the perspective of network lifetime maximization and propose a new solution, called LB-MAC (Lifetime-Balanced MAC), to achieve this goal via balancing the nodal lifetime.

## 1.1 Contributions

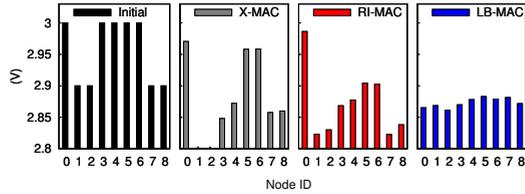
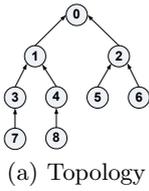
LB-MAC emphasizes the collaboration between nodes to benefit the network as a whole, even at the expense of a single node. The key idea is to allow a sensor node to adjust its MAC-layer behaviors via a few tunable parameters. The adjustment occurs in pairs between neighboring nodes; that is, each pair of neighboring sensor nodes adjust their MAC-layer behaviors together once communications between them occur.

In LB-MAC, by tuning the operational parameters, sensor nodes may operate with different *channel checking periods*, *idle listening periods* and *data retry intervals*, such that the rendezvous between sender and receiver nodes can be achieved and the communication overhead can be shifted between them. If the receiver finds itself with a longer expected lifetime than its sender, it shall attempt to take more communication overhead from the sender; otherwise, the receiver may disseminate communication overhead to the sender.

- **Shifting communication overhead from sender to receiver:** in LB-MAC, to reduce sender’s energy consumption on communication, receiver may increase its channel checking period so that sender can choose a longer retry interval while the rendezvous between sender and receiver is still guaranteed.
- **Shifting communication overhead from receiver to sender:** to save energy at the receiver side, sender may attempt data transmissions more frequently with a shorter retry interval so that receiver can shorten its channel checking period and save its energy consumption on communication. Sender may also choose to keep listening idly upon a data arrival and the rendezvous between sender and receiver is then triggered solely by receiver’s periodic beacons.

This way, the minimal nodal lifetime among communicating neighbors can be extended. As a result, the network lifetime may be prolonged.

We have implemented LB-MAC in TinyOS and evaluated it on a sensor network testbed. Experimental results show that, comparing with state-of-the-art MAC protocols such as X-MAC [3], RI-MAC [4] and SEESAW [5], LB-MAC achieves the design goal of significantly prolonging the sensor network lifetime through balancing nodal lifetime. Figure 1 shows the sample results with a simple tree topology; nodes 5, 6, 7, 8 are source nodes and the data rate is two packets per second. As shown in Figure 1(b), after 1.4 hours of network operation, sensor nodes running X-MAC or RI-MAC all experience severe imbalance in their nodal residual energy. In comparison, with the proposed LB-MAC protocol, all nodes have almost the same level of residual energy despite the unbalanced initial



(b) Initial and residual voltage levels after the network operates for 1.4 hours with X-MAC, RI-MAC and LB-MAC, respectively. Note that, nodes 1 and 2 are bottleneck nodes in the network.

**Fig. 1.** A simple comparison of the proposed LB-MAC scheme with two state-of-the-art MAC protocols

nodal energy levels. As a result, the sensor network may be able to operate for a much longer time till all sensor nodes’ batteries are depleted at approximately the same time. More evaluation results can be found in Section 5.

### 1.2 Related Work

Among MAC protocols proposed for duty cycle sensor networks [3, 4, 6, 7], S-MAC [7] and T-MAC [8] are representative synchronous protocols that require neighboring nodes to be time-synchronized and thus can align the active and sleep intervals of neighbor nodes to wake up only during the common active time periods. As the active periods usually are short, substantial energy can be saved. However, strictly synchronizing the clocks of neighboring nodes may impose high overhead.

B-MAC [6] and X-MAC [3] are representative sender-initiated asynchronous MAC protocols. In B-MAC, the rendezvous between sender and receiver is established through long preambles initiated by the sender, and X-MAC improves B-MAC by replacing the long preamble with a sequence of short, strobed preambles. A node running X-MAC may stop sending short preambles upon receiving an EarlyACK from its target receiver, thus saving more energy than B-MAC. To work under a wider range of traffic conditions, RI-MAC [4] and A-MAC [9] adopt a receiver-initiated beacon-based strategy. Each node wakes up periodically and sends out a short beacon to explicitly notify its neighbors that it is ready to receive data. When a node has data to transmit, it wakes up and waits for a beacon from its receiver. Once such a beacon is received, it starts sending the data. Compared to the sender-initiated preamble-based protocols, a receiver-initiated protocol only requires receiver to keep radio on for a short period after sending a beacon and therefore saves the receiving energy cost. Additionally, the receiver-initiated nature allows efficient collision resolution which can effectively save the transmission energy cost when channel contention is severe.

Different from [3,4,6,9], where the MAC parameters are predetermined before deployment and usually the same on all nodes in the network, MAC parameter

tuning in duty cycle sensor networks has been studied in [5, 10–12]. In [11], a controller is implemented on individual sensor nodes to dynamically adjust the radio duty cycle based on network traffic condition where no collaboration exists between neighboring nodes. Authors in [12] proposed a protocol to reduce radio duty cycles by scheduling rendezvous between neighbor nodes based on the relative end-to-end delay requirement and the network traffic condition. Though these works efficiently reduce individual nodal energy consumption, they may not improve the network lifetime in general. ZeroCal [10] is a MAC layer protocol which adaptively tunes the wakeup intervals between sender and receiver to balance the energy consumption of them; however, the proposed scheme does not guarantee the end-to-end delay bound as the wakeup interval can be extended to save nodal energy. Additionally, ZeroCal does not consider the adaptation of other MAC parameters such as channel checking interval and data retry interval, which can further prolong the network lifetime with proper tuning. SEESAW [5] was proposed to balance the energy consumption between sender and receiver through adapting the data retry interval at the sender side and the channel checking period at the receiver side. Though SEESAW yields longer network lifetime than B-MAC and S-MAC, the effectiveness of SEESAW is limited by several factors. Firstly, as a sender-initiated only protocol, SEESAW mandates a minimum channel checking period at the receiver side, which may incur unnecessary energy consumption under light traffic conditions. Secondly, the policies used in SEESAW for balancing nodal lifetime are empirical and not adaptive to varying network conditions.

### 1.3 Organization

The rest of the paper is organized as follows. Section 2 presents analytical preliminaries. Section 3 describes the design of the proposed LB-MAC protocol, which is followed by its implementation details in TinyOS in Section 4. Experimental results are presented in Section 5, and Section 6 concludes the paper.

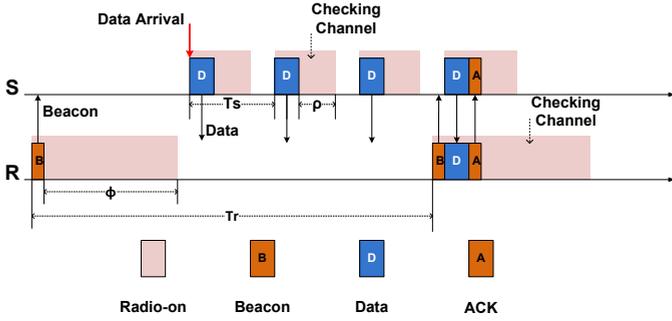
## 2 Preliminaries

In this section, we first define a generic model for duty cycle MAC protocols. Based on this model, an analytical study is conducted to provide a theoretical foundation for the design of LB-MAC protocol.

### 2.1 A Generic Model for Duty Cycle MAC Protocols

The behaviors of sensor nodes in a duty cycle MAC protocol can be generalized as follows:

- As a receiver, a sensor node periodically wakes up to interact with potential senders. During each wakeup, the sensor node can (i) check the channel activity for incoming messages, or (ii) send out alive notifications to waiting senders, or (iii) perform both.



**Fig. 2.** A generic model for duty cycle MAC protocols

- As a sender, a sensor node turns on radio immediately after a new data packet arrives. To deliver the data packet, the sensor node can (i) send a data packet to the target receiver and wait for ACK, or (ii) wait for target receiver’s alive notification to start data transmission, or (iii) perform both.

A sensor node may participate in the network activity as sender, receiver or both at the same time.

As the data packet transmission time is relatively small and can be in the same fold as a preamble in many sensor network applications, both LPL scheme in TinyOS 2.1 and UPMA-XMAC [13] protocol use data packets to replace the preambles. Similarly, in our analysis and design, we also let senders send data packets instead of preambles. Figure 2 illustrates the behaviors of sensor nodes in a generic duty cycle MAC protocol and Table 1 lists the six main parameters to characterize a MAC protocol.

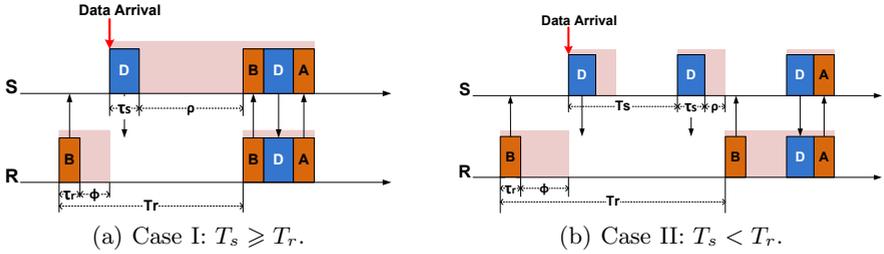
**Table 1.** MAC protocol parameters

$T_s$	sender’s data retry interval
$\rho$	sender’s idle listening period
$\tau_s$	transmission duration of a data packet
$T_r$	receiver’s wakeup interval
$\phi$	receiver’s channel checking period
$\tau_r$	transmission duration of a beacon

The above generic model can be instantiated to a certain MAC protocol with proper assignments to the parameters. For example, as shown in Table 2, the X-MAC protocol can be obtained by setting  $\tau_r = 0$  (i.e., receiver does not send any beacon),  $\tau_s = \tau$  (which is the data transmission duration),  $T_s = \epsilon$  (which is the sum of  $\tau$  and the ACK turnaround time),  $\rho = T_s - \tau_s$ , and  $\phi = 20\text{ms}$ . RI-MAC [4] can be obtained by setting  $\tau_r = \tau$ ,  $\tau_s = 0$  (i.e., sender silently waits for receiver’s beacon without sending a data packet),  $T_s = \infty$ ,  $\rho = T_s - \tau_s = \infty$

**Table 2.** Existing MAC protocol settings

Protocol	$T_s$	$\tau_s$	$\rho$	$T_r$	$\tau_r$	$\phi$
RI-MAC	$\infty$	0	$T_s - \tau_s$	user defined	$\tau$	7ms [4]
A-MAC	$\infty$	0	$T_s - \tau_s$	user defined	$\tau$	128 $\mu$ s [9]
X-MAC	$\epsilon$	$\tau$	$T_s - \tau_s$	user defined	0	20ms [3]
SEESAW	$\frac{\phi}{1.2}$	$\tau$	$\epsilon - \tau$	user defined	0	dynamic [5]


**Fig. 3.** Rendezvous between sender and receiver

(i.e., sender keeps listening idly as long as it has packets to send), and  $\phi = 7$ ms. Note that,  $T_r$  is shown as “user defined” for all four protocols listed in the table, as it is typically specified by the user to satisfy certain delay requirement.

## 2.2 Relation between Parameters

Without loss of generality, the behaviors of a sender and a receiver shown in Figure 2 are as follows.

- When a new data packet arrives, the sender sends out the packet and monitors the channel. If a receiver’s beacon is received, it retransmits the data packet; if an ACK is received, it stops the transmission and goes back to sleep; if neither ACK nor beacon is received within  $\rho$  time, it goes back to sleep. After it has slept for  $(T_s - \tau_s - \rho)$  time, the sender wakes up again and repeats the above procedure.
- The receiver wakes up every  $T_r$  interval, sends out a beacon, and monitors the channel for  $\phi$  time. If a data packet is received within  $\phi$  time, it replies with an ACK; otherwise, it goes back to sleep.

In order to deliver a data packet within user defined one-hop delay  $T_r$ , the relations between  $T_s$ ,  $T_r$ ,  $\rho$  and  $\phi$  shall satisfy certain conditions, as detailed in the following two cases:

- Case I:  $T_s \geq T_r$ . Consider a scenario where a sender fails in its first transmission attempt of a data packet because the target receiver is asleep, as shown

in Figure 3(a). If the sender goes to sleep before the receiver wakes up, it will retransmit the packet at  $T_s$  time after the first transmission attempt; that is, a delay of at least  $T_s$  will be incurred, which is longer than  $T_r$ . Therefore, to ensure that the data packet is delivered within delay  $T_r$ , the sender shall instead remain awake until the receiver wakes up to send a beacon, and then retransmit the packet. In other words, the following condition must be satisfied:

$$\rho \geq T_r - \tau_s - \tau_r - \phi. \tag{1}$$

- Case II:  $T_s < T_r$ . In this case, as shown in Figure 3(b), if a sender fails in its first transmission attempt of a data packet because the target receiver is asleep, it does not need to remain awake to wait for the receiver’s beacon. Instead, the sender can go back to sleep and wake up later every  $T_s$  time as long as it is guaranteed that, the sender’s awake durations overlap with the receiver’s very next awake duration. Specifically, the following condition must be satisfied:

$$T_s \leq \phi + \tau_s + \tau_r + \rho. \tag{2}$$

Using the parameters listed in Table 2, it is easy to verify that Inequality (1) holds for RI-MAC and A-MAC, and Inequality (2) holds for X-MAC and SEE-SAW. In our design, we require either Inequality (1) or (2) to hold. Note that, in the above analysis, packet collisions or losses are not considered for simplicity; but they are considered in our protocol design and implementation as elaborated in Sections 3 and 4.

### 2.3 Lifetime Estimation

Based on the above analysis, the expected lifetime of a pair of sender and receiver, denoted as  $L_s$  and  $L_r$  respectively, can be estimated as follows:

$$L_s = \frac{e_s}{\frac{T_{r,r}}{2} \cdot \frac{(\rho_s + \tau_s)}{T_{s,s}} \cdot r_s \cdot P + g_s}, \tag{3}$$

and

$$L_r = \frac{e_r}{\frac{(\tau_r + \phi_r)}{T_{r,r}} \cdot P + g_r}, \tag{4}$$

where  $e_s$  and  $e_r$  are the amount of residual energy at sender and receiver respectively,  $r_s$  is the outgoing data rate at the sender,  $P$  is the amount of energy consumed when a node’s radio is on per unit of time, and  $g_s$  and  $g_r$  are energy consumption rates of sender and receiver for other causes.

In the above estimation, the data packet outgoing rate is assumed to be low such that there is no queueing packet to be sent, which is typical in many low duty cycle applications [14, 15]. To send each data packet, the sender needs to wait for about  $\frac{T_{r,r}}{2}$  time on average, with the radio duty cycle value of  $\frac{\rho_s + \tau_s}{T_{s,s}}$ . Therefore, it consumes about  $\frac{T_{r,r}}{2} \cdot \frac{\rho_s + \tau_s}{T_{s,s}} \cdot P$  energy to deliver a packet on average.

As for the receiver, it wakes up for  $\tau_r + \phi_r$  time every  $T_{r,r}$  interval. Hence, its energy consumption rate can be estimated as  $\frac{\tau_r + \phi_r}{T_{r,r}} \cdot P$ .

More generally, a sensor node  $i$  may act as both a sender and a receiver in the network, and its expected lifetime  $L_i$  can be estimated by considering its power consumption for communicating with each of its senders (similar to the analysis in Equation (3)) and each of its receivers (similar to the analysis in Equation (4)). Details are omitted due to space limitation.

## 2.4 Problem Statement and Design Principle

The goal of this work is to design a MAC protocol that maximizes the lowest nodal lifetime in the network via adjustment of MAC-layer behaviors of sensor nodes. Formally, it can be described as follows:

**Given:**

$$P, \{\tau_{s,i}\}, \{\tau_{r,i}\}, \{T_{r,i,j} \text{ where } j \text{ is any sender of } i\}, \\ \{r_{i,j} \text{ where } j \text{ is any receiver of } i\}, \{g_i\}, \{e_i\}.$$

**Objective:**

$$\max \min\{L_i\}$$

**Subject to: for any sender-receiver pair  $(i, j)$ ,**

$$T_{s,i,j} \leq \phi_{j,i} + \tau_{s,i} + \tau_{r,j} + \rho_{i,j}, \text{ or} \\ \rho_{i,j} \geq T_{r,j,i} - \tau_{s,i} - \tau_{r,j} - \phi_{j,i},$$

**Output:**

$$\{T_{s,i,j} \text{ where } j \text{ is any receiver of } i\}, \\ \{\rho_{i,j} \text{ where } j \text{ is any receiver of } i\}, \{\phi_{i,j} \text{ where } j \text{ is any sender of } i\}.$$

Directly solving this optimization problem is impractical because it requires each node to know the residual energy levels, energy consumption rates and data arrival rates of all other nodes in the network. Acquiring these information could incur high communication overhead because of potentially large network scale and dynamic nature of the information. So instead, we approach the problem in a distributed, localized and low-cost manner. Specifically, each node only periodically coordinates with its neighboring nodes to balance the lifetime between them:

- If a node as a receiver finds itself with longer expected lifetime than its sender, it shall attempt to take more communication overhead from the sender. According to Equation (3), this can be done by increasing  $T_s$  and/or decreasing  $\rho$  at the sender side, accompanied with increasing  $\phi$  at the receiver side to satisfy Inequality (1) or (2).
- On the other hand, if a receiver finds itself with shorter expected lifetime than its sender, it shall attempt to shift more communication overhead to the sender. According to Equation (4), this can be done by decreasing  $\phi$  at the receiver side, accompanied with decreasing  $T_s$  and/or increasing  $\rho$  at the sender side to satisfy Inequality (1) or (2).

This way, the minimal nodal lifetime among communicating neighbors can be extended. As a result, the network lifetime may be prolonged.

### 3 LB-MAC Design

In LB-MAC, each pair of sender and receiver adapt their MAC-layer behaviors through tuning a few operational parameters:  $T_s$  (data retry interval) and  $\rho$  (idle listening period) for the sender and  $\phi$  (channel checking period) for the receiver. The receiver acts in a leading role. Based on the lifetime information piggybacked in data packets from the sender, the receiver decides a proper  $\phi$  and piggybacks it in the ACK to the sender. The receiver’s behavior is elaborated in Section 3.1. Upon receiving an ACK, the sender extracts the piggybacked  $\phi$ , based on which it adjusts its parameters  $T_s$  and  $\rho$  as elaborated in Section 3.2.

#### 3.1 Receiver’s Behavior

The operational flowchart of an LB-MAC node as receiver is shown in Figure 4. Every  $T_r$  interval (i.e., when the wakeup timer is fired), the receiver turns on radio, sends a beacon and keeps monitoring the channel for  $\phi$  time. During the monitoring period, if a data packet is received, the following information will be extracted from the packet: *sender’s estimated lifetime* and *sender’s outgoing data rate*. The receiver may adjust its  $\phi$  according to the information, and return the updated  $\phi$  to the sender in the ACK.

When a receiver adjusts its  $\phi$ , the adjustment scale shall be small. This is because multiple sender-receiver pairs may adjust their parameters concurrently; each pair makes the adjustment according to their current knowledge of their energy consumption rates, which can be affected by the adjustments made by

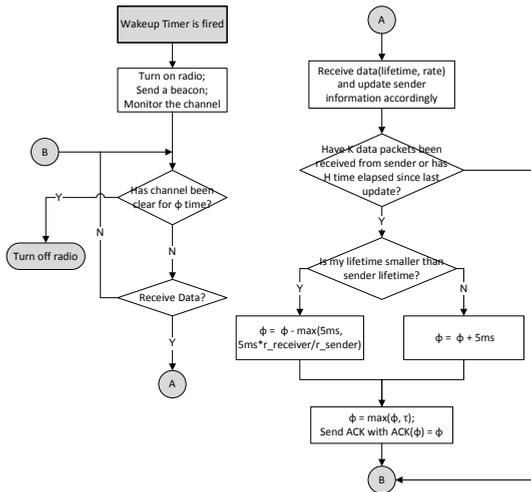


Fig. 4. Receiver’s behavior in LB-MAC

other pairs. Hence, if the adjustment scale is too large, thrashing may occur and energy may be wasted. Specifically, the receiver behaves as follows, which is also shown in the flowchart:

- $\phi$  may be updated only when some threshold conditions are satisfied: (i) the receiver has received at least  $K$  data packets; or (ii) at least  $H$  seconds have elapsed since the last time when  $\phi$  was updated. In our design and implementation, we empirically choose  $K=30$  and  $H=60$ , when the difference between the two nodes' lifetime may change since the previous update of  $\phi$  and a new update shall be necessary.
- When the receiver has a longer lifetime than the sender, it increases  $\phi$  by 5 ms; otherwise, it decreases  $\phi$  by the amount of  $\max(5\text{ms}, \frac{r_{receiver}}{r_{sender}} \cdot 5\text{ms})$ , where  $r_{receiver}$  and  $r_{sender}$  are the outgoing data rates of receiver and sender, respectively. This way, the receiver's energy consumption rate could be reduced quickly; this design is based on the consideration that data collection is usually the major communication pattern in a sensor network and hence a receiver usually serves multiple senders concurrently.

### 3.2 Sender's Behavior

The operational flowchart of an LB-MAC node as sender is shown in Figure 5. When a data packet arrives at the sender, the buffer is checked first. If the buffer is not empty, the data packet is simply put into the buffer. Otherwise, the sender makes the first attempt of transmitting this packet, and meanwhile starting a timer that will expire every  $T_s$  time. Then, if the transmission succeeds, the sender goes to sleep; otherwise, it remains awake for  $\rho$  time and then goes to sleep. The sender also attempts transmitting data packets when it receives a beacon from the target receiver. In addition, the sender wakes up and attempts transmitting packets whenever the data retry timer expires. The timer is turned off when the buffer becomes empty.

When the sender receives an ACK to its data packet, it will adjust  $T_s$  and  $\rho$  according to the  $\phi$  value carried in the ACK, such that (i) either Inequality (1) or (2) is satisfied to ensure data packet delivery within delay  $T_r$ , and (ii) its idle listening time is as short as possible. The adjustment is performed as follows:

- If Inequality (1) is satisfied, the average radio-on time of the sender for the successful delivery of each data packet is estimated as

$$\begin{aligned}
 & \tau_s + \frac{\rho}{2} \\
 & \geq \tau_s + \frac{T_r - \tau_s - \tau_r - \phi}{2}, \text{ due to Inequality (1)} \\
 & = \frac{T_r + \tau_s - \tau_r - \phi}{2}.
 \end{aligned}$$

That is, the minimum average radio-on time is

$$\frac{T_r + \tau_s - \tau_r - \phi}{2}, \tag{5}$$

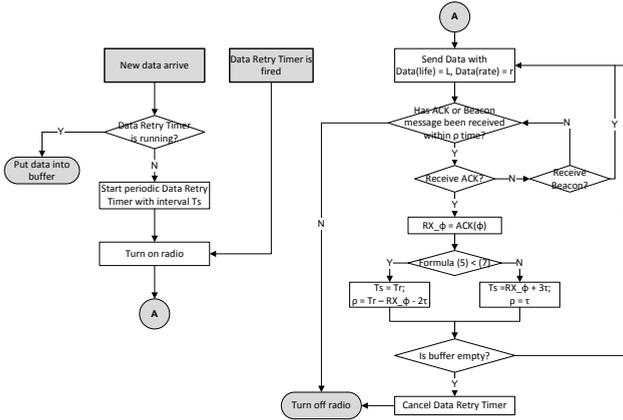


Fig. 5. Sender’s behavior in LB-MAC

which can be achieved as long as

$$T_s = T_r, \text{ and } \rho = T_r - \tau_s - \tau_r - \phi. \tag{6}$$

- If Inequality (2) is satisfied, the average radio-on time for the successful delivery of a data packet is estimated as

$$\begin{aligned} & \frac{1}{2} \cdot \frac{T_r}{T_s} \cdot (\tau_s + \rho) \\ & \geq \frac{T_r \cdot (\tau_s + \rho)}{2(\phi + \tau_s + \tau_r + \rho)}, \text{ due to Inequality (2)} \\ & = \frac{T_r}{2} \cdot \left(1 - \frac{\phi + \tau_r}{\phi + \tau_r + \tau_s + \rho}\right). \end{aligned}$$

As the transmission duration of a software ACK is similar to the transmission duration of a beacon, the minimum  $\rho$  value is  $\tau_r$ . Therefore, the minimum average radio-on time is

$$\frac{T_r}{2} \cdot \left(1 - \frac{\phi + \tau_r}{\phi + \tau_r + \tau_s + \tau_r}\right), \tag{7}$$

which is achieved when

$$T_s = \phi + \tau_r + \tau_s + \tau_r, \text{ and } \rho = \tau_r. \tag{8}$$

- If the sender’s radio-on time computed by Formula (5) is less than or equal to that computed by Formula (7),  $T_s$  and  $\tau_s$  are set as in Equation (6) to reduce sender’s energy consumption; otherwise,  $T_s$  and  $\tau_s$  are set as in Equation (8).

### 3.3 Robustness of the LB-MAC Design

One of the key features in LB-MAC is that, LB-MAC allows neighboring nodes to exchange additional control information and then adjust their MAC-layer

behaviors together. Therefore, in order for LB-MAC to be practically useful, it is critical to ensure that LB-MAC functions properly in the presence of packet losses, route changes and multiple concurrent senders, all of which are inevitable in practical environments.

**Loss of Data Packets.** Loss of data packets has no effects on LB-MAC. The sender will keep retransmitting till the data is delivered successfully, or till the maximum retry limit has been reached and data is discarded. During the process, both sender and receiver operate with the previously-agreed upon MAC parameters.

**Loss of ACK Packets.** Loss of an ACK may cause sender and receiver to lose synchronization of their MAC behaviors, since the important decision on MAC behavior adaptation may be piggybacked in the ACK. For example, a receiver may decide to reduce  $\phi$  and carry this decision in an ACK. Unfortunately, due to loss of this ACK, the sender never gets notified of the change and continues operating with a  $T_s$  value that is larger than the new  $\phi$ . As a result, the Inequality (1) or (2) given in Section 2.2 may be violated, and the rendezvous between sender and receiver is lost. To deal with this issue, LB-MAC adopts a rescue mechanism. The idea is to allow a sender to change  $\rho$  to  $T_r$  when the number of data retries exceeds  $\lceil T_r/T_s \rceil$ , which is the maximum number of data retries during a  $T_r$  interval. This is to guarantee that sender and receiver recover from the loss of MAC behavior synchronization in at most  $2T_r$  time.

**Handling of Channel Contention.** In LB-MAC,  $T_s$  is a tunable parameter and when it becomes too small, data messages are sent in very short intervals, which may cause severe contention to the channel and a large number of packet collisions. As a result, senders may waste energy contending for the channel. To deal with this situation, LB-MAC sets the minimum  $T_s$  value to 20ms which is specified in [3] for the same purpose.

**Handling of Multiple Concurrent Senders.** In LB-MAC, as the parameter tuning is made pair-wisely, a node who serves as a common receiver for multiple senders may decrease  $\phi$  for one sender and then lose the rendezvous with other senders. To address this problem, a receiver keeps record of the  $\phi$  value scheduled with each sender, and choose the largest  $\phi$  value as its own channel checking period such that the rendezvous with all senders can be satisfied.

**Handling of Route Changes.** In practice, a sender node may switch to a new receiver due to routing updates. The receiver may waste energy on unnecessarily long channel checking period if it keeps using the  $\phi$  value scheduled for the stale sender. In LB-MAC, the receiver node periodically checks and drops stale senders and thus the  $\phi$  value for staled senders will not be used. Similarly, a sender node may also drop stale receivers periodically if they don't interact with each other after certain period. When the sender switches to a new receiver, it sets  $\rho = T_r$  to wait for the target receiver's beacon and establish the rendezvous.

## 4 LB-MAC Implementation

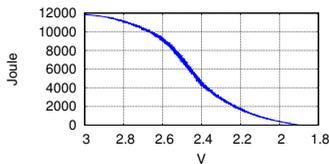
LB-MAC has been implemented in TinyOS 2.1.0. The core scheduling component of LB-MAC is the LBMACScheduler component, which resides atop the radio core layer and handles all operations of message processing and parameter tuning based on the flow chart as shown in Figures 4 and 5. Some adaption code of LB-MAC is also developed for the radio core layer which provides a variety of low-level supports for the LBMACScheduler component. In the following, we present the message formats defined by LB-MAC and discuss some implementation issues.

In LB-MAC, the beacon message is used by a receiver either as a beacon sent upon its wakeup or as a software ACK sent to acknowledge the reception of a data packet. The type field defined in a general TOS message header is reused in the beacon messages. The same as in the implementation of RI-MAC, the type field in a beacon message carries the backoff window size decided by the receiver, to allow its senders to choose their backoff slots based on this value. Different from RI-MAC and A-MAC, LB-MAC adds 2-byte fields to each beacon message for carrying parameters  $\phi$ . In a data packet, estimated nodal lifetime and outgoing data rate of the sender are piggybacked to the end of the data payload, both of which are used by the receiver when deciding MAC parameters as discussed in Section 3.1.

In LB-MAC, estimated nodal lifetime is a key factor in determining the MAC mode and parameters, and the capability to measure nodal residual energy level is necessary for lifetime estimation. Though the residual energy level can be estimated using energy meter devices [16] or monitored precisely with integrated software and hardware support [17], only a few existing sensor motes [18,19] are designed with the required hardware support. To cope with this constraint, the implementation of LB-MAC adopts a software based residual energy estimation scheme, which has also been used in existing works [20].

The basic idea is that, for each type of battery, the mapping between a battery voltage reading and the residual energy level is found, and the information is then input to our LB-MAC module. As an example, Figure 6 shows the mapping that we have found for a pack of two AA Ni-Mn batteries. Particularly, we find the mapping using the following method: A sensor mote is configured to start working when its battery voltage reading is 3 V, in full duty cycle with sensing activity enabled. Every a small time interval, the voltage reading is recorded and time-stamped, until the energy is depleted. Let these records be  $(t_i, v_i)$  for  $i = 0, \dots, n$ , where  $t_0 = 0$ ,  $v_0 = 3$  and  $t_n$  is the time when the energy is found depleted, and let the initial energy of the batteries be estimated as  $e_0 = t_n \cdot v_0 \cdot I$ , where  $I$  is the current. Then, the residual energy level corresponding to battery reading  $v_i$  can be estimated as  $e_i = e_0 \cdot \frac{t_n - t_i}{t_n}$ .

With the input mapping between battery readings and residual energy levels, a sensor node can estimate its residual nodal lifetime based on its voltage reading and consumption rate. To reduce the overhead for storing the mapping information, the whole voltage range can be divided into multiple segments such that the mapping relation for each segment can be captured with a simple function; this



**Fig. 6.** Mapping between battery voltage readings and residual energy levels (for two AA Ni-Mn batteries)

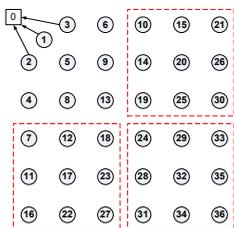
way, only the functions and segment information need to be stored [21]. Though our current implementation requires extra measurement work to find out the mapping between battery readings and residual energy levels, we believe that more precise and pervasive hardware support will be available in near future due to the increasing demand of lifetime-aware services [1].

### 5 Performance Evaluation

Experiments have been conducted to evaluate the performance of LB-MAC and compare it with X-MAC, RI-MAC and SEESAW, in terms of network lifetime, average per-hop delay, data delivery ratio and network power consumption. Here, the network power consumption is defined as the amount of energy consumed in the whole network divided by the network lifetime.

The testbed is composed of 37 TelosB motes, organized as a  $6 \times 6$  grid (shown in Figure 7). Node 0 is connected to PC, keeping its radio on all the time to serve as the sink. Two scenarios of event-driven sensing and data collection are emulated:

- Static events scenario, in which static events are assumed to be detected by sensors 28, 29, 32, 34, 35 and 36 only. These sensors (i.e., source nodes) generate data packets at a certain rate and forward them hop by hop to the sink.
- Dynamic events scenario, in which dynamic events are emulated to occur and be detected by sensors in one of the three dotted rectangles at a time. A sensor that detects an event generates data packets at a certain rate and forward them hop by hop to the sink.



**Fig. 7.** Deployment of sensor nodes in testbed experiments. The routing pathes are determined by the CTP protocol.

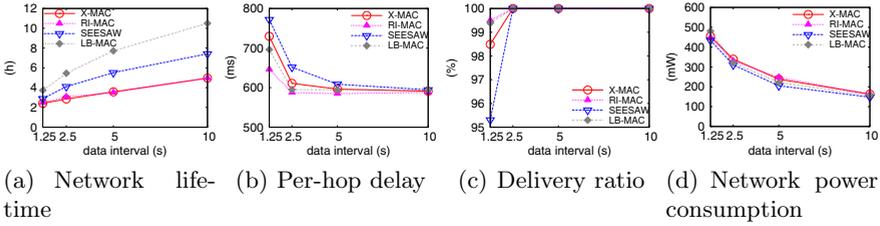


Fig. 8. Performance comparison with static events and uniform initial nodal energy

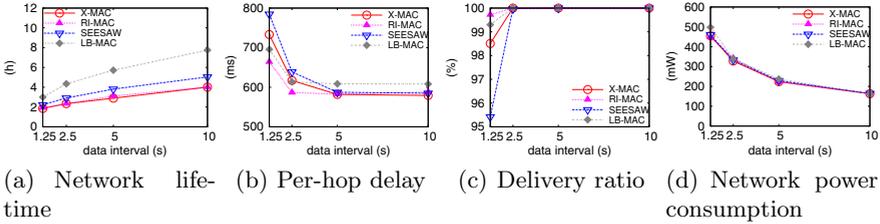


Fig. 9. Performance comparison with static events and non-uniform initial nodal energy

In the experiments, the CTP [22] protocol is used to form and adjust the paths for data packet forwarding, and the recorded longest hop count in the network is 8. Parameter  $T_r$  is fixed at 1 second for all protocols, and parameters  $\phi$ ,  $\rho$  and  $T_s$  for X-MAC, RI-MAC, and SEESAW are set as in Table 2 which adopt the default settings in [3], [4] and [5].

Due to the prolonged network lifetime, it may take weeks to drain completely fully-charged batteries of sensor nodes. In order to conduct the experiments in a reasonable amount of time while demonstrating the feature and performance of evaluated protocols, we study how fast a sensor node consumes a designated small portion of its full energy, and evaluate its nodal lifetime as the time period during which this designated amount of energy is consumed. Particularly, at the beginning of each experiment, the initial available nodal energy distribution may be uniform or non-uniform. When the distribution is uniform, each sensor node’s initial available energy is designated to 400 Joules; when it is non-uniform, the initial available energy varies between 300 and 400 Joules.

### 5.1 Performance with Static Events

We first evaluated the performance of LB-MAC with static events when the initial energy distribution is uniform or non-uniform. The results are shown in Figures 8 and 9.

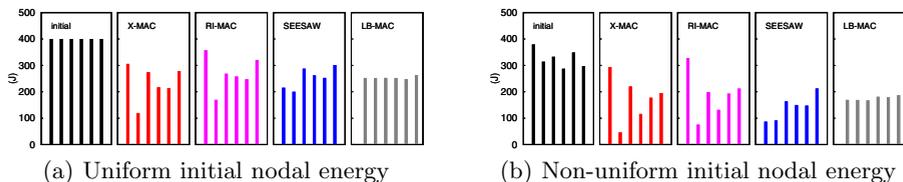
As shown in Figures 8(a) and 9(a), LB-MAC achieves longer network lifetime than RI-MAC, X-MAC and SEESAW do under various conditions. Particularly, when the data generation interval is 2.5 seconds, the network lifetime achieved by

LB-MAC is about 90% longer than that achieved by RI-MAC and X-MAC, and 30% longer than that achieved by SEESAW, with uniform initial nodal energy. The leading edge is even more significant (e.g., about 50% longer than SEESAW) with non-uniform initial nodal energy. This is mainly due to the following reasons. As RI-MAC and X-MAC fix MAC protocol parameters, bottleneck nodes have the heaviest workloads, consume more energy than others, and their nodal lifetime constrains the network lifetime as shown by Figure 10. Instead, LB-MAC dynamically adjusts MAC parameters to shift the communication overhead, balance nodal lifetime as shown by Figure 10, and hence significantly increases the network lifetime. SEESAW also attempts to balance nodal lifetime, but its capability of parameter adjustment is less effective than LB-MAC because its adjustments follow a set of fixed policies that are not adaptive to changes in conditions.

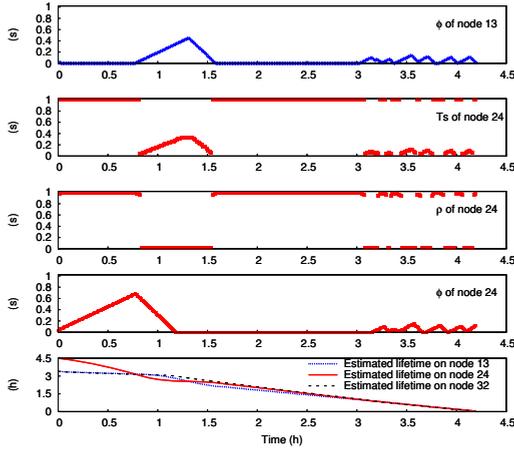
The evaluation results in (b), (c) and (d) of Figures 8 and 9 show that, LB-MAC does not compromise its performance in other aspects to attain longer network lifetime. Specifically, LB-MAC maintains similar packet delivery ratio, per-hop delay and network power consumption as RI-MAC, X-MAC and SEESAW.

**A Working Trace.** To further illustrate how LB-MAC adaptively changes MAC behaviors to balance nodal lifetime, Figure 11 plots changing traces of parameters  $T_s$ ,  $\rho$  and  $\phi$  at forwarding node 24, as well as the changing trace of  $\phi$  at node 13, on path  $32 \rightarrow 24 \rightarrow 13$ .

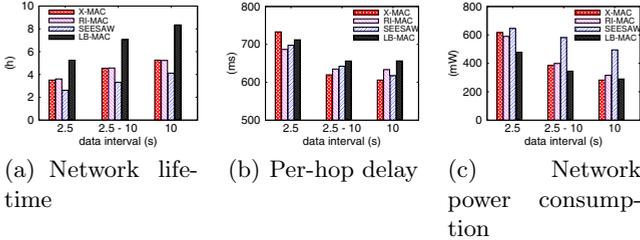
At time instance 0, the estimated nodal lifetime of node 24 is higher than that of nodes 13 and 32. Hence, as the receiver of link  $32 \rightarrow 24$ , node 24 increases its  $\phi$  to shift the communication overhead from 32 to itself. Meanwhile, as the receiver of link  $24 \rightarrow 13$ , node 13 assigns a very small value to its  $\phi$  such that it keeps awake for only a very short duration every time it wakes up. In response, node 24 sets its  $T_s$  and  $\rho$  as in Equation (6); that is, whenever node 24 has data packets to send to node 13, it keeps awake to wait for beacon from node 13 and then transmit the packets. This way, workload associated with communication between nodes 24 and 13 is mostly shifted to node 24. As a result of the above parameter adjustment, the nodal lifetime of nodes 13, 24 and 32 is balanced gradually during the time interval  $[0, 0.8\text{h}]$ . Shortly after time instance 0.8h,



**Fig. 10.** Snapshot of available remaining energy of nodes 1, 5, 13, 24, 32 and 36 after 2 hours of network operation with static events (data generation interval at source nodes is 2.5 seconds)



**Fig. 11.** Changing traces of  $\phi$ ,  $\rho$  and  $T_s$  at node 24 and  $\phi$  at node 13 on path  $32 \rightarrow 24 \rightarrow 13$



**Fig. 12.** Performance comparison with dynamic events and non-uniform initial nodal energy (note: data interval “2.5-10” means data packets are generated at an interval uniformly distributed in [2.5s, 10s])

node 24’s nodal lifetime drops to be shorter than that of nodes 13 and 32. Adapting to the change, node 24 decreases its  $\phi$  to shift some communication overhead to its sender node 32 on link  $32 \rightarrow 24$ , and meanwhile decreases its  $\rho$  and adjusts its  $T_s$  accordingly to shift some communication overhead to node 13 on the link  $24 \rightarrow 13$ .

### 5.2 Performance with Dynamic Events

As shown in Figure 12, with dynamic events, LB-MAC can still achieve significantly longer network lifetime than the state-of-the-art MAC protocols while maintaining similar network power consumption and per-hop delay as those protocols (packet delivery ratio for all protocols is close to 100% and the figures are omitted here). The results well demonstrate the robustness and good performance of LB-MAC in practical scenarios where (i) the nodal initial energy is heterogeneous, (ii) the routing paths and the traffic pattern are time-varying and (iii) the data sources are temporally and spatially dynamic. In particular,

the adaptiveness and performance stability of LB-MAC under different network settings are even obvious when compared to the SEESAW protocol. In the prior experiments with static events, SEESAW attains longer network lifetime and lower network power consumption than X-MAC and RI-MAC do; however, with more dynamic routing and more time-varying traffics, the performance of SEESAW is degraded significantly due to its fixed and empirical polities for MAC parameter tuning. On the contrast, LB-MAC's adaptive design enables itself to overcome these practical problems and deliver good performance stably.

## 6 Conclusion and Future Work

In this paper, we present a new MAC protocol, called LB-MAC (Lifetime-Balanced MAC), which is designed from the perspective of network lifetime maximization. The key idea of LB-MAC is to allow sensor nodes to adjust their MAC-layer behaviors dynamically so as to extend the network lifetime through balancing nodal lifetime between communicating neighbors. The effectiveness of the proposed scheme is demonstrated via in-depth experimental results. Future work will be conducted along the following directions. As many schemes have been proposed at other layers to balance nodal lifetime or nodal energy consumption, we plan to compare LB-MAC with these schemes and study the advantage and limitations of each approach. Based on the study, we will explore the feasibility and strategy of the balancing techniques through cross-layer integration with middle layer [23], routing layer [24] or services in other network layers. We will also extend LB-MAC by adding lifetime balancing support for broadcast or multicast data services. In duty cycle sensor networks, the basic approach for broadcast or multicast is to transmit data to the destination nodes through unicast one by one [13, 25] and the advanced scheme is to delegate data transmissions to different nodes [26] so the original broadcast or multicast initiator can go to sleep earlier to save energy. Such extensions are also applicable to LB-MAC, and we believe that LB-MAC's performance may be improved further if the traffic pattern (unicast, multicast or broadcast) can be used when adjusting the MAC-layer parameters.

**Acknowledgement.** This work was supported partly by the NSF under Grants CNS-0831874 and ECCS-1128312.

## References

1. Challen, G.W., Waterman, J., Welsh, M.: IDEA: Integrated Distributed Energy Awareness for Wireless Sensor Networks. In: *MobiSys* (2010)
2. Wang, W., Srinivasan, V., Chua, K.: Using Mobile Relays to Prolong the Lifetime of Wireless Sensor Networks. In: *MobiCom* (2005)
3. Buettner, M., Yee, G., Anderson, E., Han, R.: X-MAC: A Short Preamble MAC Protocol For Duty-Cycled Wireless. Sensor Networks. In: *SenSys* (2006)

4. Sun, Y., Gurewitz, O., Johnson, D.: RI-MAC: A Receiver-Initiated Asynchronous. Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks. In: SenSys (2008)
5. Braynard, R., Silberstein, A., Ellis, C.S.: Extending Network Lifetime Using an Automatically Tuned Energy-Aware MAC Protocol. In: Römer, K., Karl, H., Mattern, F. (eds.) EWSN 2006. LNCS, vol. 3868, pp. 244–259. Springer, Heidelberg (2006)
6. Polastre, J., Hill, J., Culler, D.: Versatile Low Power Media Access for Wireless Sensor Networks. In: SenSys (2004)
7. Ye, W., Heidemann, J., Estrin, D.: An Energy-Efficient MAC protocol for Wireless Sensor Networks. In: INFOCOM (2002)
8. Dam, T., Langendoen, K.: An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. In: SenSys (2003)
9. Dutta, P., Dawson-Haggerty, S., Chen, Y., Liang, C.J.M., Terzis, A.: Design and Evaluation of a Versatile and Efficient. Receiver-Initiated Link Layer for Low-Power Wireless. In: SenSys (2010)
10. Meier, A., Woehrle, M., Zimmerling, M., Thiele, L.: ZeroCal: Automatic MAC Protocol Calibration. In: Rajaraman, R., Moscibroda, T., Dunkels, A., Scaglione, A. (eds.) DCOSS 2010. LNCS, vol. 6131, pp. 31–44. Springer, Heidelberg (2010)
11. Merlin, C.J., Heinzelman, W.B.: Duty Cycle Control for Low-Power-Listening MAC Protocols. *IEEE Transactions on Mobile Computing* 9, 1508–1521 (2010)
12. Peng, Y., Li, Z., Qiao, D., Zhang, W.: Delay-Bounded MAC with Minimal Idle Listening for Sensor Networks. In: INFOCOM (2011)
13. Washington University St. Louis: UPMA Package: Unified Power Management Architecture for Wireless Sensor Networks, <http://tinocvs.sourceforge.net/tinyos/tinyos-2.x/contrib/wustl/upma/>
14. Gu, Y., He, T.: Data Forwarding in Extremely Low Duty-Cycle Sensor Networks with Unreliable Communication Links. In: SenSys (2007)
15. Wang, X., Wang, X., Xing, G., Yao, Y.: Dynamic Duty Cycle Control for End-to-End Delay Guarantees in Wireless Sensor Networks. In: IWQOS (2010)
16. Dutta, P., Feldmeier, M., Paradiso, J., Culler, D.: Energy Metering for Free: Augmenting Switching Regulators for Real-Time Monitoring. In: IPSN (2008)
17. Fonseca, R., Dutta, P., Levis, P., Stoica, I.: Quanto: Tracking Energy in Networked Embedded Systems. In: OSDI (2008)
18. Lin, K., Yu, J., Hsu, J., Zahedi, S., Lee, D., Friedman, J., Kansal, A., Raghunathan, V., Srivastava, M.: Heliomote: Enabling Long-Lived Sensor Networks. Through Solar Energy Harvesting. In: SenSys (2005)
19. Lu, G., De, D., Xu, M., Song, W.Z., Cao, J.: TelosW: Enabling ultra-low power wake-on sensor network. In: Seventh International Conference on Networked Sensing Systems, INSS (2010)
20. Lachenmann, A., Herrmann, K., Rothermel, K., Marrón, P.J.: Meeting Lifetime Goals and Providing Constant Application Quality. *ACM Trans. Sen. Netw.* 5, 36:1–36:36 (2009)
21. Zhu, T., Zhong, Z., Gu, Y., He, T., Zhang, Z.L.: Leakage-Aware Energy Synchronization for Wireless Sensor Networks. In: MobiSys (2009)
22. Gnawali, O., Fonseca, R., Jamieson, K., Moss, D., Levis, P.: Collection Tree Protocol. In: SenSys (2009)

23. Li, Z., Peng, Y., Qiao, D., Zhang, W.: LBA: Lifetime Balanced Data Aggregation in Low Duty Cycle Sensor Networks. In: INFOCOM (2012)
24. Li, Z., Peng, Y., Zhang, W., Qiao, D.: J-RoC: A Joint Routing and Charging scheme to prolong sensor network lifetime. In: ICNP (2011)
25. Klues, K., Hackmann, G., Chipara, O., Lu, C.: A Component Based Architecture for Power-Efficient Media Access Control in Wireless Sensor Networks. In: SenSys (2007)
26. Sun, Y., Gurewitz, O., Du, S., Tang, L., Johnson, D.B.: ADB: An Efficient Multi-hop Broadcast Protocol Based on Asynchronous Duty-Cycling in Wireless Sensor Networks. In: SenSys (2009)