A Lifetime-Balancing MAC Protocol under the End-to-End Delay Requirement

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

Abstract: **This article presents lifetime-balancing medium access control (LB-MAC), a new medium access control (MAC) protocol with embedded adaptivity 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 under a certain end-to-end delay requirement. It achieves this goal by dynamically tuning a comprehensive set of MAC parameters. LB-MAC is a distributed, lightweight, and scalable solution, 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 yield a significantly longer network lifetime than state-of-the-art MAC protocols such as X-MAC, RI-MAC, and SEESAW, while meeting the end-to-end delay requirement, and maintaining comparable levels of data delivery ratio and average nodal power consumption.**

Index Terms: **Lifetime-balancing, low duty cycle, medium access control (MAC), wireless sensor networks.**

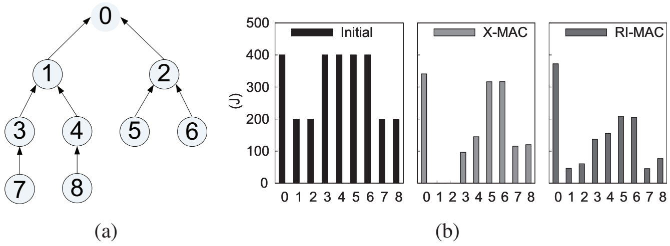


Fig. 1. The energy bottleneck effect with two state-of-the-art MAC protocols. The data generation rate is 2 packets/s and the wakeup interval is 1 s for all protocols in the experiment: (a) A tree topology where nodes 5–8 are source nodes and (b) initial and residual nodal energy after the network has operated for 1.4 h with X-MAC and RI-MAC, respectively. Note that, nodes 1 and 2 are bottleneck nodes in the network.

I. 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]–[4], 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 in a tree topology network, may cause network disconnection and 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 more 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 [5], [6] or system factors. For example, nodes with a battery of poorer quality or solar-powered nodes deployed at shady locales may have a shorter lifetime than their peers. Therefore, to maximize the network lifetime, it is important to extend the shortest nodal lifetime among all sensor nodes.

Despite the need for a holistic approach to address the energy conservation challenge and prolong the network lifetime, most of the current research on medium access control (MAC) protocol design has focused on reducing the energy consumption and extending the operational lifetime of individual sensor nodes. For example, as shown in Fig. 1, when sensor nodes run X-MAC [7] or RI-MAC [8] - two state-of-the-art MAC protocols for sensor networks - they experience severe imbalance in residual nodal energy after 1.4 h of network operation. As a result, the network lifetime is limited due to such energy bottleneck effect.

To remedy this deficiency, we investigate the MAC protocol design from the perspective of network lifetime maximization and propose a new solution, called lifetime-balancing MAC (LB-MAC), to achieve this goal via balancing the nodal lifetime between neighboring sensor nodes. We have implemented LB-MAC in TinyOS and experiment results show that LB-MAC outperforms the state-of-the-art MAC protocols in terms of network lifetime while meeting the end-to-end delay requirement, and maintaining comparable levels of data delivery ratio and average nodal power consumption.

LB-MAC emphasizes collaboration between sensor nodes to benefit the network as a whole, even at the expense of a single node. The key idea is that neighboring nodes adjust their MAC-layer behaviors together (only when there are data communications between them) via the following tunable parameters: *Wakeup interval* and *channel checking period* at the receiver side, and *data retry interval* and *idle listening period* at the sender side. These parameters are tuned carefully in a certain manner so that (i) the rendezvous between sender and receiver can be guaranteed; (ii) the incurred communication overhead (for rendezvous maintenance) can be shifted between them; and

This paper is specially handled by EICs and Division Editors with the help of three anonymous reviewers in a fast manner.

Yang Peng is with the University of Washington Bothell, 18115 Campus Way NE, Bothell, Washington 98011, USA, email: yangpeng@uw.edu.

Zi Li is with the LinkedIn Corporation, Mountain View, California 94043, USA, email: lizi860929@gmail.com.

Wensheng Zhang and Daji Qiao are with Iowa State University, Ames, Iowa 50011, USA, email: {wzhang, daji}@iastate.edu.

Digital object identifier: 10.1109/JCN.2017.000008

(iii) the end-to-end delivery delay satisfies the requirement. This way, the node with a shorter expected lifetime than its communicating neighbor can extend its lifetime by shifting more communication overhead to the neighbor. As a result, the network lifetime may be prolonged gradually. The behaviors of LB-MAC can be summarized as follows.

- **Shifting communication overhead from a sender to a receiver.** If a receiver finds itself with a longer expected lifetime than its sender, it may decrease the wakeup interval or increase the channel checking period, which allows the sender to choose a longer data retry interval (to reduce its communication energy consumption) while the rendezvous between the sender and the receiver can still be guaranteed.
- **Shifting communication overhead from a receiver to a sender.** On the other hand, to save energy at the receiver side, the sender may attempt data transmissions more frequently (with a shorter data retry interval) so that the receiver can increase the wakeup interval or shorten the channel checking period to reduce its communication energy consumption. The sender may even choose to keep listening idly upon a data arrival; this way, receiver can reduce the channel checking period to minimal, and the rendezvous between the sender and the receiver is triggered solely by the receiver's periodic beacons.

The rest of this paper is organized as follows. Section II discusses the state-of-the-art MAC protocols for duty cycle sensor networks. Section III presents the analytical preliminaries and the problem statement. Section IV describes the design of the proposed LB-MAC protocol, which is followed by its implementation details in Section V. Experimental results are presented in Section VI and Section VII concludes the paper.

II. RELATED WORK

A. Fixed Duty Cycle MAC Protocols

For many duty cycle MAC protocols, the MAC operational parameters are predetermined before deployment for simplicity of usage and implementation, and the parameter settings are usually the same for all nodes in the network.

Among these protocols, B-MAC [9] and X-MAC [7] are representative sender-initiated asynchronous MAC protocols. In B-MAC, the rendezvous between a sender and a receiver is established through long preambles initiated by the sender. X-MAC improves over 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.

As B-MAC and X-MAC are optimized mainly for light traffic conditions, the preambles may congest the channel and block data transmissions in the scenarios of busty or high traffic load. To work under a wider range of traffic conditions, RI-MAC [8] and A-MAC [10] 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 the target 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 a 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 the channel contention is severe. However, it is worth noting that under very light traffic, the receiver-initiated protocols may incur higher energy cost than the sender-initiated protocols due to the overhead of sending receiver's beacons and waiting for incoming traffic.

B. Dynamic Duty Cycle MAC Protocols

Different from fixed duty cycle MAC protocols, MAC parameter tuning in duty cycle sensor networks has also been studied in [11]–[20]. In particular, SEESAW [11] was proposed to balance the energy consumption between a sender and a receiver through adapting the data retry interval at the sender side and the channel checking period at the receiver side. Though SEESAW yields a longer network lifetime than B-MAC and S-MAC, the effectiveness of SEESAW is limited by several factors. Firstly, as a sender-initiated protocol, SEESAW mandates a minimum channel checking period at the receiver side, which may incur unnecessary energy consumptions. Secondly, the policies used in SEESAW for balancing nodal lifetime are empirical and not adaptive to varying network conditions. Thirdly, MAC parameters such as the wakeup interval and the idle listening period are fixed in SEESAW, which, if tuned properly, could prolong the network lifetime further.

Both DDCC [15] and CyMAC [16] target at improving individual nodal energy efficiency. In DDCC [15], a controller is implemented on individual sensor nodes to dynamically adjust the radio duty cycle based on the network traffic condition. CyMAC [16] was proposed to reduce radio duty cycle by scheduling rendezvous between neighboring nodes based on the relative end-to-end delay requirement and the network traffic condition. Though these schemes may reduce individual nodal energy consumption, they may not effectively improve the network lifetime due to the lack of collaboration between nodes. MaxMAC [21] is a MAC protocol that can adapt between X-MAC and pure CSMA mode of operations given different network traffic conditions to deal with the tradeoff between energy-efficiency and throughput/delay. More recently, AEDP [22] was proposed to dynamically adjust the radio CCA threshold to improve network reliability and duty cycle based on application-specified bounds. Although these protocols can improve nodal energy-efficiency, and deal with the exposed throughput or latency drawbacks of duty cycle MAC protocols, they may not be able to improve the network lifetime as a whole.

ZeroCal [14] is a MAC layer protocol which adaptively tunes the wakeup intervals between a sender and a receiver to balance their energy consumption; however, the proposed scheme cannot preserve the end-to-end delay as the wakeup interval may be extended indefinitely to save nodal energy. In addition, ZeroCal does not consider the adjustment of other MAC parameters such as channel checking period and data retry interval, which, if tuned properly, could further prolong the network lifetime. GDSIC [17] is another work that targets at improving the fairness of energy utilization in duty cycle sensor networks. It proposes a similar idea as in ZeroCal by dynamically tuning the nodal wakeup interval. Different from ZeroCal, GDSIC decides the individual nodal wakeup interval through solving distributed convex optimization problems. Though the network lifetime can be prolonged in GDSIC, the desired end-to-end delay may not be preserved after parameter tuning.

pTunes [18] is a recent work that adjusts the MAC parameters dynamically for low-power sensor networks. It formalizes three optimization problems, in each of which the network lifetime, the end-to-end reliability, or the end-to-end latency is the optimization objective while the other two are the optimization constraints, and the MAC-layer parameters including radio-on duration, radio-off duration, and the number of retransmission attempts are the output. Furthermore, pTunes is a centralized solution that requires periodic network state collection and parameter dissemination. Hence, it may not be feasible in practice.

C. Uniqueness of the Proposed LB-MAC Protocol

Different from existing works, our proposed LB-MAC protocol aims to improve the network lifetime while preserving the end-to-end delivery delay. It achieves this goal with a unique approach that adjusts nodal radio duty cycles (i) collaboratively between neighbors, and (ii) systematically via a comprehensive set of tunable operational parameters at the MAC layer. It is a distributed, lightweight, and scalable solution as the control information is only exchanged locally between neighbors.

D. Techniques beyond MAC Layer

Energy-aware routing protocols [3], [23], [24] have been proposed to prolong the sensor network lifetime. Recently, the authors in [1], [25], [26] proposed specially-designed energy-aware routing schemes for duty cycle sensor networks. In all these works, the main idea is to route packets through nodes with a higher residual energy or a longer nodal lifetime such that nodes with a lower energy or a shorter lifetime can participate less in data transmission activities. As a result, the minimum nodal lifetime in the network may be extended and the network lifetime may be prolonged. In addition, approaches to prolong the network lifetime through cross-layer design have been proposed in [27]–[31]. In these works, [27] attempts to maximize the network lifetime via joint routing and MAC design, [31] solves the problem via joint routing and congestion control, and [28] tackles the problem through joint optimal design of physical, MAC, and routing layers in time slotted networks. LB-MAC is complementary to these schemes and can be integrated with them to further improve network lifetime.

III. ANALYSIS AND PROBLEM STATEMENT

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

A. Duty Cycle MAC Protocols: A Generic Model

Fig. 2 illustrates the behaviors of sensor nodes with a generic duty cycle MAC protocol, which are explained below. Table 1 lists the parameters to characterize a MAC protocol.

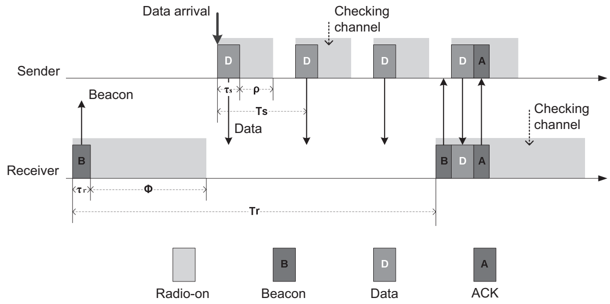


Fig. 2. A generic model for duty cycle MAC protocols.

Table 1. MAC protocol parameters.

T_s	Sender's data retry interval
ρ	Sender's idle listening period
η_s	Boolean value: 1 - sender sends a probe; 0 - no
T_r	Receiver's wakeup interval
ϕ	Receiver's channel checking period
η_r	Boolean value: 1 - receiver sends a beacon; 0 - no
τ_s	Transmission duration of a probe
τ_r	Transmission duration of a beacon

As a receiver, a sensor node wakes up every T_r interval to interact with potential senders. At the beginning of each wakeup, the sensor node may send out a beacon message to waiting senders (the transmission duration of the beacon message is τ_r), or silently wait for its senders to transmit packets. During the wakeup period, the sensor node checks the channel activity for ϕ time for incoming messages. If a data packet is received within ϕ time, it replies with an ACK; otherwise, it goes back to sleep.

On the other hand, when a sensor node has a data packet to send, it wakes up every T_s interval to interact with the target receiver. At the beginning of each wakeup, the sensor node may transmit the data packet[1] immediately or wait silently for the target receiver's beacon to start the data transmission. During the idle listening period ρ, if an ACK is received, the procedure ends as the data packet has been delivered successfully; if a beacon is received instead, it retransmits the data packet; if neither ACK nor beacon is received, it goes back to sleep and wakes up at the next T_s interval and to repeat the above procedure. Note that, a sensor node may participate in the network activity as a sender, a receiver, or both at the same time.

The above model can be instantiated to a specific MAC protocol by assigning proper values to the parameters. For example, as shown in Table 2, the X-MAC [7] protocol can be obtained by setting $\eta_r = 0$ (i.e., receiver does not send any beacon), $\eta_s = 1$, $T_s = \epsilon$, $\rho = T_s - \eta_s\tau_s$, and $\phi = 20$ ms. ϵ is the sum of τ_s, *tx-rx turnaround time* and duration of an ACK transmission, which is the minimum radio-on time when a node sends a data packet. RI-MAC [8] can be obtained by setting $\eta_r = 1$, $\eta_s = 0$ (i.e., sender waits silently for receiver's beacon without sending a data packet), $T_s = \infty$, $\rho = \infty$ (i.e., sender keeps listening idly as long as it has packets to send[2]), and $\phi = 7$ ms (a platform

[1] As the data packet transmission time is usually small and can be in the same fold as a probe in many sensor network applications, the LPL scheme in TinyOS 2.1 [32] uses data packets to replace preambles. Similarly, in our design and analysis, we also let senders send data packets instead of probes.

[2] In practice, after a certain period, the MAC layer may stop the retries, turn

Table 2. MAC protocol settings.

	T_s	η_s	ρ	T_r	η_r	ϕ
RI-MAC	∞	0	∞	fixed	1	7 ms
A-MAC	∞	0	∞	fixed	1	128 μs
X-MAC	ϵ	1	$\epsilon - \tau_s$	fixed	0	20 ms
SEESAW	$\phi/1.2$	1	$\epsilon - \tau_s$	fixed	0	dynamic
ZeroCal	ϵ	1	$\epsilon - \tau_s$	dynamic	0	fixed
GDSIC	∞	0	∞	dynamic	1	fixed
AutoSync	ϵ	1	$\epsilon - \tau_s$	dynamic	0	fixed
MaxMAC	ϵ	1	$\epsilon - \tau_s$	dynamic	0	fixed
LB-MAC	dynamic	1	dynamic	dynamic	1	dynamic

dependent value).

B. Analysis of Rendezvous Condition and Delivery Delay

Though the rendezvous condition for existing MAC protocols has already been analyzed in related work as discussed in Section II, we include the analysis of rendezvous condition (based on the generic model given in Section III-A) for completeness.

To ensure that sender and receiver meet within T_r time to deliver a data packet, the MAC protocol parameters shall satisfy the following condition, called the *rendezvous condition*:

$$(\eta_r \tau_r + \phi) + (\eta_s \tau_s + \rho) > \min\{T_s, T_r\}, \quad (1)$$

which is summarized from the following cases:

- Case I: $0 < T_s \leqslant T_r$. In this case, as shown in Fig. 3(a), if a sender fails in its first transmission attempt of a data packet (because the target receiver is asleep), it goes back to sleep and wakes up later. To ensure that sender and receiver meet within T_r time, one of the sender's future awake durations shall overlap with the receiver's very next awake duration. That is, the following condition shall be satisfied:

$$(\eta_r \tau_r + \phi) > T_s - (\eta_s \tau_s + \rho), \quad (2)$$

  which is equivalent to:

$$(\eta_r \tau_r + \phi) + (\eta_s \tau_s + \rho) > T_s = \min\{T_s, T_r\}. \quad (3)$$

- Case II: $T_s > T_r$. In this case, sender's data retry interval is longer than receiver's wakeup interval (e.g., in RI-MAC and A-MAC, $T_s = \infty$ as sender simply waits silently for receiver's beacon to start the data transmission). In order to deliver a data packet within T_r time, sender needs to keep listening the channel till the receiver's very next beacon is received, as illustrated in Fig. 3(b). Therefore, the following condition shall be satisfied:

$$(\eta_s \tau_s + \rho) > T_r - (\eta_r \tau_r + \phi), \quad (4)$$

  which is equivalent to:

$$(\eta_r \tau_r + \phi) + (\eta_s \tau_s + \rho) > T_r = \min\{T_s, T_r\}. \quad (5)$$

It is easy to verify that rendezvous condition (1) holds for all existing MAC protocols, including sender-initiated protocols such as X-MAC and SEESAW, and receiver-initiated protocols

off the radio, and ask the upper layer whether to keep the radio on to wait for the receiver.

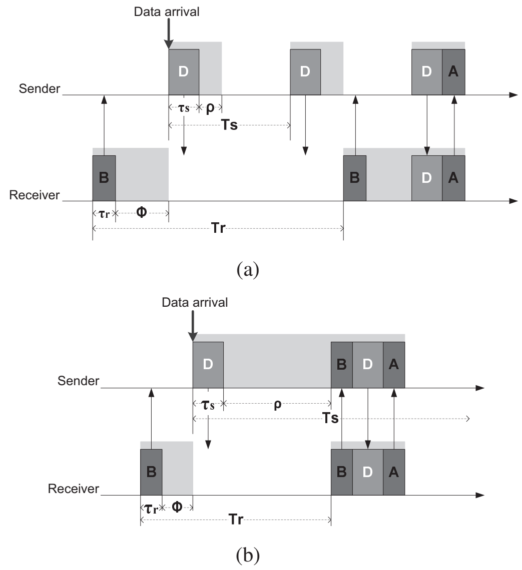


Fig. 3. Rendezvous between sender and receiver: (a) Case I: $0 < T_s \leqslant T_r$ and (b) case II: $T_s > T_r$.

such as RI-MAC and A-MAC. When designing LB-MAC, we also require the condition to hold. In fact, we require a slightly more stringent rendezvous condition:

$$\phi + \rho \geqslant \min\{T_s, T_r\}, \quad (6)$$

which simplifies the design and analysis of the protocol by omitting the small values of τ_s and τ_r.

When the rendezvous condition is satisfied, the maximal one-hop delay from node x to node y under a perfect channel condition is:

$$D_{x \to y} = T_r(y) - \phi(y). \quad (7)$$

This is also the maximal time that our proposed LB-MAC protocol allows a sensor node to hold a data packet (upon its generation or reception) before informing the upper layer of a delivery failure. Subsequently, the maximal end-to-end delivery delay of a flow f under a perfect channel condition is:

$$D_f = \sum_{\text{all hops on flow} f} D_{x \to y}. \quad (8)$$

C. Analysis of Nodal Lifetime

Based on the above analysis, the expected lifetime of sender x and receiver y under a perfect channel condition, denoted as $L_s(x)$ and $L_r(y)$ respectively, can be estimated as follows:

$$L_s(x) = \frac{e(x)}{D_{x \to y} \frac{\rho(x)}{T_s(x)} R(x)\, P + g(x)} \quad (9)$$

and

$$L_r(y) = \frac{e(y)}{\frac{\phi(y)}{T_r(y)} P + g(y)}, \quad (10)$$

where (i) $e(x)$ and $e(y)$ are the amount of residual energy at sender and receiver, respectively, (ii) $R(x)$ is the sender's outgoing data rate, (iii) P is the amount of energy consumed when a node's radio is on for one unit of time (transmission and reception power are assumed to be the same [8], [33], [34]), and (iv) $g(x)$ and $g(y)$ are the energy consumption rates of sender and receiver, respectively, for other causes.

In the above estimation, the sender's outgoing data rate is assumed to be low so that there is no queueing at the sensor nodes, which is typical in low duty cycle sensor network applications [25], [35], [36]. Therefore, to send a data packet, sender x needs to wait for $D_{x\to y}$ time with a radio duty cycle of $\rho(x)/T_s(x)$. As a result, it consumes $D_{x\to y}(\rho(x)/T_s(x))\cdot R(x)P$ power for data transmissions. For receiver y, it wakes up for $\phi(y)$ time every $T_r(y)$ interval. Hence, its energy consumption rate for receiving can be estimated as $(\phi(y)/T_r(y))P$.

As a sensor node may act as both sender and receiver in the network, its expected lifetime shall be estimated by considering its power consumption for communicating with each of its senders and each of its receivers by combining (9) and (10). From the equations, we can see that the nodal lifetime of sender and receiver can be balanced through tuning their MAC layer parameters (i.e., T_s, ρ, T_r, and ϕ) collaboratively.

D. Problem Statement

To effectively prolong the sensor network lifetime, ideally, all sensor nodes shall work together to maximize the minimal nodal lifetime in the entire network. Unfortunately, it is impractical to solve this optimization problem in a realistic sensor network, because it requires each node to know the following information of every other node in the network: the residual nodal energy, the energy consumption rate, and the data arrival rate. Acquiring this information could incur very high communication overhead because of the potentially large network scale and the dynamic nature of the information. So instead, we study the following localized problem when communication occurs between two nodes from j to i:

Objective:

- $\max\min\{L(i), L(j)\}$, where $L(i)$ and $L(j)$ are i's and j's nodal lifetime (as defined in Section III-C).

Subject to:

- *Rendezvous condition:* $\phi(i,j)+\rho(j,i)\geqslant\min\{T_s(j,i),T_r(i,j)\}$.
- *End-to-end delay requirement:* for each flow f that link $j \rightarrow i$ belongs to, $\sum_{\text{all hops on flow } f} D_{x\to y}\leqslant D_{e2e}$, where $D_{x\to y}$ is defined in (7) and D_{e2e} is the end-to-end delay requirement.

Output:

- For node i,
  - its $T_r(i,j)$ and $\phi(i,j)$ values to communicate with j;
- For node j,
  - its $T_s(j,i)$ and $\rho(j,i)$ values to communicate with i;
  - its $T_r(j,k)$ and $\phi(j,k)$ values to communicate with its own sender node k.

The goal of this problem is to maximize the minimal lifetime between a sender and a receiver. As such procedure occurs in all communication pairs, the minimal nodal lifetime in the entire network, i.e., the network lifetime, may be improved gradually.

E. Design Principle

We propose a protocol called LB-MAC to address the problem defined above. In LB-MAC, coordination only take place locally between a communication pair, which adjust their MAC-layer parameters together in a collaborative manner, as follows.

If a receiver finds itself with a longer expected lifetime than its sender, it shall attempt to shift more communication overhead from the sender. According to (9) and (10), this can be done by increasing ϕ and/or decreasing T_r at the receiver side, accompanied with increasing T_s and/or decreasing ρ at the sender side, as long as both rendezvous condition and delay preservation requirement are satisfied.

On the other hand, if a receiver finds itself with a shorter expected lifetime than its sender, it shall attempt to shift more communication overhead to the sender via decreasing ϕ and/or increasing T_r at the receiver side, and decreasing T_s and/or increasing ρ at the sender side.

IV. THE PROPOSED LB-MAC PROTOCOL

In LB-MAC, whenever there are data communications between a pair of sensor nodes, they adapt their MAC-layer behaviors together via piggybacking information in the data/ACK exchanged between them. For example, based on the information piggybacked in a data packet from a sender, the receiver decides its T_r and ϕ values and embeds them in an ACK to the sender. Upon reception of the ACK, the sender adjusts its T_s and ρ values accordingly to ensure that the rendezvous condition is satisfied. In LB-MAC, the receiver takes a leading role to coordinate the MAC behaviors of itself and each sender. This way, senders don't need to exchange information between themselves to adjust their behaviors, thus saving more energy. Receiver's and sender's behaviors are elaborated in Sections IV-A and IV-B, respectively, where we use flow $x'\rightarrow x\rightarrow y\rightarrow y'$ as an illustrative example to explain the behavioral details.

A. Receiver's Behavior

The operational flowchart of an LB-MAC node as a receiver is shown in Fig. 4. Every T_r interval (i.e., when the wakeup timer is fired), receiver y turns on radio, sends a beacon, and monitors the channel for ϕ time. During the monitoring period, if a data packet is received from sender x, the following information will be extracted from the data packet: *x's estimated nodal lifetime*, *one-hop communication delay from x's previous-hop node x' to x – denoted as $D_{x'\to x}$*, and *x's tuning credit – denoted as $D_{\text{credit}}(x)$*.

Here, node x's *tuning credit* refers to the cumulative delay savings generated by receiver y's previous adjustments of T_r and ϕ values. For example, if receiver y increases its ϕ or decreases its T_r for 100 ms, according to (7), the one-hop communication delay from x to y is decreased by 100 ms, and hence the tuning credit is increased by 100 ms. Such a *tuning credit* may be spent by x (or y) later if it needs to adjust its operational parameters to prolong the nodal lifetime but at the expense of increased one-hop communication delay from x' to x (or from x to y), without increasing the end-to-end delay. If receiver y

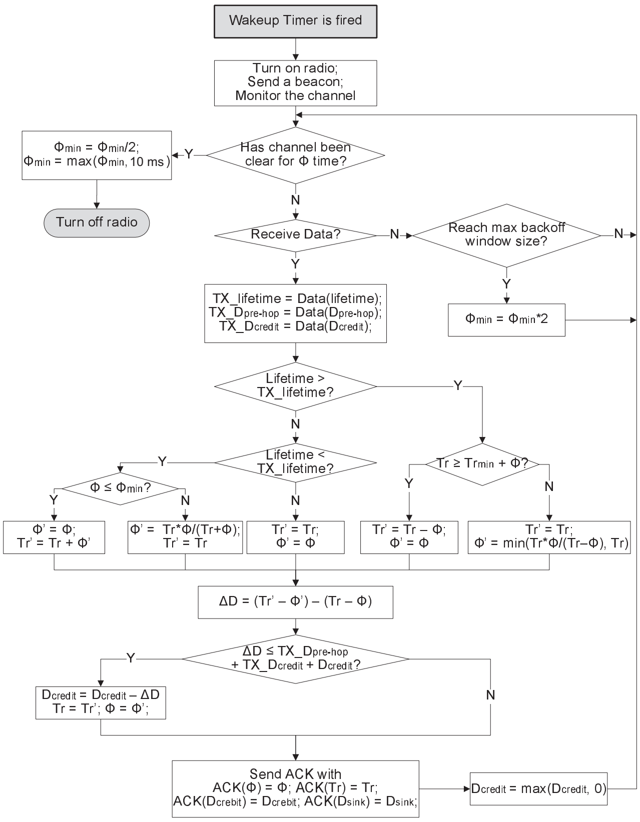


Fig. 4. Receiver's behavior in LB-MAC.

keeps increasing ϕ or decreasing T_r, the tuning credit can be accumulatively increased over period. Initial value of the tuning credit is zero. In Section IV-B, we will give examples in Fig. 6 to explain how the tuning credit may be utilized by either a receiver or a sender.

When a receiver adjusts its operational parameters, the sender needs to adjust its own operational parameters accordingly to ensure that the rendezvous condition is satisfied. As a result, the nodal lifetime of both sender and receiver, as well as the one-hop communication delay between them, may be affected, which have been analyzed in Section III. Therefore, receiver y is allowed to adjust its operational parameters (T_r and ϕ) only if the parameter adjustment does not cause the end-to-end delay to increase. This can be guaranteed as long as the following condition is satisfied:

$$\Delta D_{x\to y} \leqslant D_{x'\to x} + D_{\text{credit}}(x) + D_{\text{credit}}(y), \tag{11}$$

where $D_{\text{credit}}(x)$ and $D_{\text{credit}}(y)$ represent the tuning credits of nodes x and y respectively, and $\Delta D_{x\to y}$ is the increased one-hop communication delay from x to y as a result of y's parameter adjustment. $\Delta D_{x\to y}$ can be calculated as follows:

$$\Delta D_{x\to y} = D^{\text{new}}_{x\to y} - D_{x\to y} = (T'_r - \phi') - (T_r - \phi). \tag{12}$$

Condition (11) implies that the maximum increment allowed in $D_{x\to y}$ (without increasing the end-to-end delay) is $\max \Delta D = D_{x'\to x} + D_{\text{credit}}(x) + D_{\text{credit}}(y)$. As shown in (13) below, the maximum increment can be accommodated by (i) asking x to adjust its operational parameters to reduce $D_{x'\to x}$ to $D^{\text{new}}_{x'\to x} = 0$, and (ii) using up all the tuning credits saved for communication hops $x \to y$ and $y \to y'$.

$$\begin{aligned}
&D^{\text{new}}_{x'\to x\to y\to y'} \\
&= D^{\text{new}}_{x'\to x} + D^{\text{new}}_{x\to y} + D^{\text{new}}_{y\to y'} \\
&= 0 + [D_{x\to y} + \max \Delta D_{x\to y}] + D_{y\to y'} \\
&= 0 + [D_{x\to y} + D_{x'\to x} + D_{\text{credit}}(x) + D_{\text{credit}}(y)] + D_{y\to y'} \\
&= D_{x'\to x} + [D_{x\to y} + D_{\text{credit}}(x)] + [D_{y\to y'} + D_{\text{credit}}(y)] \\
&\leqslant [D_{x'\to x} + D_{\text{credit}}(x')] + [D_{x\to y} + D_{\text{credit}}(x)] \\
&\quad + [D_{y\to y'} + D_{\text{credit}}(y)] \\
&= D^{\text{current}}_{x'\to x\to y\to y'}.
\end{aligned} \tag{13}$$

As shown in the middle of Fig. 4, receiver y attempts to adjust T_r and ϕ according to the following rules, and the adjustment takes effect only when the resulting $\Delta D_{x\to y}$ satisfies the delay requirement (11).

- When the receiver has a longer expected lifetime than the sender, it decreases T_r gradually in steps of ϕ till T_r reaches a default minimal value then it starts to increase ϕ to $\frac{T_r}{T_r-\phi}\phi$ iteratively till $\phi = T_r$.
- When the receiver has a shorter expected lifetime, it decreases ϕ to $\frac{T_r}{T_r+\phi}\phi$ iteratively till reaching $\phi_{\min}$; then it starts to increase T_r in steps of ϕ. Here, $\phi_{\min}$ is an online parameter that we use to indicate the severity of the current channel contention; a larger $\phi_{\min}$ value corresponds to more severe channel contention. In Section IV-C.3, we will discuss in more detail how channel contention is handled in LB-MAC.

The reason for choosing such adjustment steps for T_r and ϕ is to ensure that T_r is always an integer multiple of ϕ, which simplifies the design, analysis, and implementation of LB-MAC.

After adjusting T_r and ϕ, y updates $D_{\text{credit}}(y)$ to $D_{\text{credit}}(y) = D_{\text{credit}}(y) - \Delta D_{x\to y}$, and embeds it together with the new T_r and ϕ parameters in the ACK to the sender. Note that the updated $D_{\text{credit}}(y)$ could be a negative value as the y's tuning credit alone may not be able to accommodate the delay increment caused by the new T_r and ϕ parameters; help may be needed from sender x, as we explained above in (13). In addition, y also embeds a $D_{\text{sink}}(y)$ value in the ACK, which indicates the maximum allowed end-to-end delay from y to the sink. D_{sink} has an initial value of zero, and it will be constantly updated after the communication starts. Upon receiving the ACK from receiver y, sender x adjusts its operational parameters to ensure that both conditions (6) and (11) are satisfied, which we discuss next.

B. Sender's Behavior

The operational flowchart of an LB-MAC node as a sender is shown in Fig. 5. Every T_s interval (i.e., when the data retry timer is fired), sender x turns on radio, sends a data packet, and monitors the channel for ρ time. Within ρ time, if a beacon is received, node x retransmits the data packet; on the other hand, if an ACK is received from receiver y, x extracts the following information from the ACK: $T_r(y)$, $\phi(y)$, $D_{\text{credit}}(y)$, and $D_{\text{sink}}(y)$, based on which to adjust its operational parameters as follows.

Step 1: As shown in the middle of the flowchart, to sat-

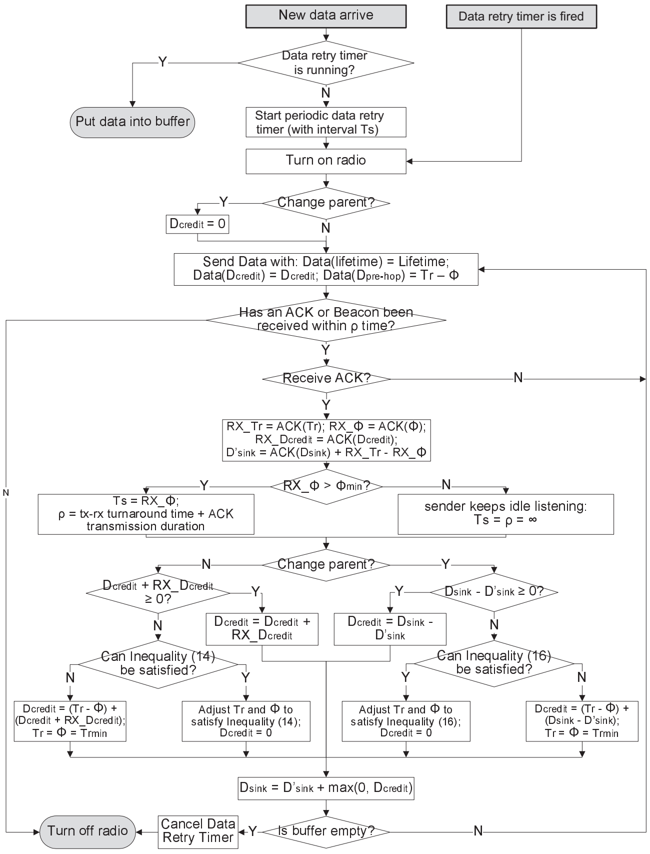


Fig. 5. Sender's behavior in LB-MAC.

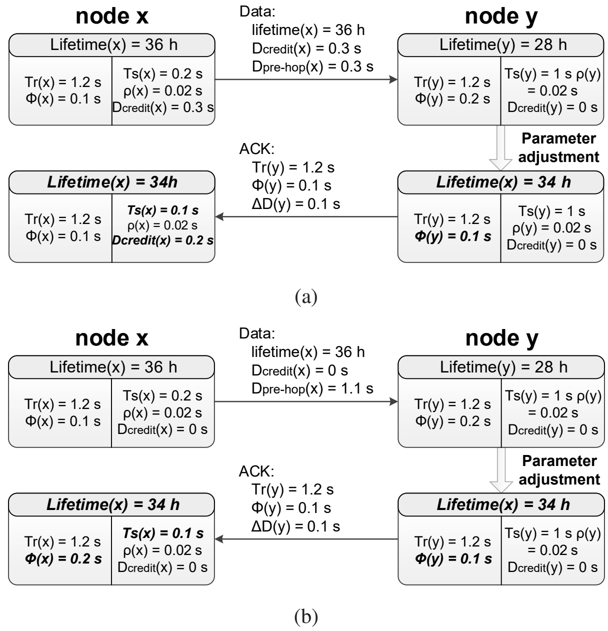


Fig. 6. Parameter tuning examples. Tuned parameters are shown in *italic bold* font: (a) As receiver y has a shorter expected lifetime than sender x, it decreases ϕ which results in an increase in one-hop communication delay: $\Delta D_{x\to y} = (1.2 - 0.1) - (1.2 - 0.2) = 0.1$ s. $D_{\text{credit}}(y)$ is updated from 0s to $D_{\text{credit}}(y) = 0 - ((1.2 - 0.1) - (1.2 - 0.2)) = -0.1$ s. Since $D_{\text{credit}}(x) + D_{\text{credit}}(y)$ is greater than 0 s, sender x simply pays off $D_{\text{credit}}(y)$ using the saved credit: $D_{\text{credit}}(x) = 0.3 - 0.1 = 0.2$ s and (b) similar to (a), receiver y decreases ϕ which results in an increase of 0.1 s in one-hop communication delay and $D_{\text{credit}}(y)$ is updated from 0 s to -0.1 s. This time, however, since $D_{\text{credit}}(x) + D_{\text{credit}}(y)$ is less than 0 s, sender x has to adjust its own ϕ value so that $D_{x'\to x}$ is decreased to offset $D_{\text{credit}}(y)$ and end-to-end delay remains the same.

isfy rendezvous condition (6), x sets $T_s(x)$ to $\phi(y)$, and $\rho(x)$ to the sum of *tx-rx turnaround time* and the duration of an ACK transmission, except when $\phi(y)$ is less than $\phi_{\min}$. In the latter situation, x remains awake and keeps listening idly for beacon or ACK from receiver y by setting $T_s(x) = \rho(x) = \infty$ (similar to how RI-MAC operates), instead of retransmitting data every short $\phi(y)$ time.

Step 2a: As shown in the bottom left of the flowchart, if $D_{\text{credit}}(x) + D_{\text{credit}}(y) < 0$, this means that the saved tuning credit won't be able to pay off the remaining delay increment that receiver y demands. In this situation, x needs to adjust its own T_r and ϕ parameters (used to communicate with its own sender x') to satisfy delay requirement (11). Specifically, x will first decrease T_r and then increase ϕ, if needed, till it finds the first pair of T_r and ϕ that satisfy the following inequality:

$$(D_{x'\to x} - D^{\text{new}}_{x'\to x}) + (D_{\text{credit}}(x) + D_{\text{credit}}(y)) \geqslant 0, \quad (14)$$

where

$$D_{x'\to x} = T_r(x) - \phi(x);\ D^{\text{new}}_{x'\to x} = T'_r(x) - \phi'(x).$$

Fig. 6 gives two examples on how the sender adjusts its parameters under different scenarios.

In case Inequality (14) cannot be satisfied even after $T_r(x)$ has been decreased to its minimum: $T'_r(x) = \min T_r$, and $\phi(x)$ has been increased to its maximum: $\phi'(x) = T'_r(x) = \min T_r$, x has to ask for its senders to help accommodate the delay increment. This is accomplished via updating its tuning credit:

$$\begin{aligned} D_{\text{credit}}(x) &= (D_{x'\to x} - D^{\text{new}}_{x'\to x}) + (D_{\text{credit}}(x) + D_{\text{credit}}(y)) \\ &= (T_r(x) - \phi(x)) - (T'_r(x) - \phi'(x)) \\ &\quad + (D_{\text{credit}}(x) + D_{\text{credit}}(y)) \\ &= (T_r(x) - \phi(x)) + (D_{\text{credit}}(x) + D_{\text{credit}}(y)), \end{aligned} \quad (15)$$

which is now a negative value and will be embed in the next ACK to its own sender x'.

Note that the above adjustment may only increase ϕ and/or decrease T_r; hence, the rendezvous condition remains valid after the adjustment.

Step 2b: In practice, routes may change and a sender may switch to a different receiver to reach the sink. Let y^{new} denote the new receiver for sender x. As shown in the bottom right of the flowchart, x behaves differently depending on the relation between $D_{\text{sink}}(x)$ – the maximum allowed end-to-end delay from x to y to the sink (i.e., the old route), and $D'_{\text{sink}}(x) = D_{\text{sink}}(y^{\text{new}}) + T_r(y^{\text{new}}) - \phi(y^{\text{new}})$ which is the maximum allowed end-to-end delay from x to y^{new} to the sink (i.e., the new route).

- If $D_{\text{sink}}(x) \geqslant D'_{\text{sink}}(x)$, meaning that the end-to-end delay is reduced after the route change, x simply updates its tuning credit to: $D_{\text{credit}}(x) = D_{\text{sink}}(x) - D'_{\text{sink}}(x)$.
- If $D_{\text{sink}}(x) < D'_{\text{sink}}(x)$, this means that the new route yields a longer delay than the old route. Similar to **Step 2a**, x will

first decrease T_r and then increase ϕ, if needed, till it finds the first pair of T_r and ϕ that satisfy the following inequality:

$$(D_{x' \to x} - D^{new}_{x' \to x}) + (D_{\text{sink}}(x) - D'_{\text{sink}}(x)) \geqslant 0. \quad (16)$$

In case Inequality (16) cannot be satisfied even after $T_r(x)$ has been decreased to its minimum: $T'_r(x) = \min T_r$, and $\phi(x)$ has been increased to its maximum: $\phi'(x) = T'_r(x) = \min T_r$, x updates its tuning credit to:

$$\begin{aligned} D_{\text{credit}}(x) &= (D_{x' \to x} - D^{\text{new}}_{x' \to x}) + (D_{\text{sink}}(x) - D'_{\text{sink}}(x)) \\ &= (T_r(x) - \phi(x)) - (T'_r(x) - \phi'(x)) \\ &+ (D_{\text{sink}}(x) - D'_{\text{sink}}(x)) \\ &= (T_r(x) - \phi(x)) + (D_{\text{sink}}(x) - D'_{\text{sink}}(x)), \end{aligned} \quad (17)$$

and embeds it in the next ACK to its own sender x'. This way, the increased delay caused by route change will be gradually reduced and the end-to-end delay can be preserved.

C. Robustness of the LB-MAC Design

In order for LB-MAC to be practically useful, it is critical to ensure that LB-MAC functions properly in the presence of failed data packet transmissions, route changes, and multiple concurrent senders, all of which occur often in practical environments.

C.1 Failed Data Packet Transmission

A failed data packet transmission may be due to loss of data packet itself or loss of ACK. Though loss of data packet has no effects on rendezvous between sender and receiver in LB-MAC, loss of ACK may cause sender and receiver to lose synchronization of their MAC-layer behaviors, because the important decision on MAC behavior adaptation may be piggybacked in the ACK. For example, a receiver may decide to reduce ϕ and carry this decision in an ACK. Unfortunately, due to loss of ACK, the sender never gets notified of the change and continues to operate with a T_s value that is larger than the new ϕ. As a result, rendezvous condition (6) given in Section III-B may be violated.

LB-MAC deals with this situation as follows. The sender keeps retransmitting the data packet with the previously-agreed upon MAC-layer operational parameters till either the packet is delivered successfully or when the packet has been retried for $(T_r - \phi)$ time. In the latter case, the sender stops the retries and informs the upper layer of the delivery failure. When a future data packet targeting at the same receiver arrives, the sender remains awake to listen idly (by setting T_s and ρ to ∞) till the receiver's beacon is received to reestablish the rendezvous. This is to ensure a timely recovery from the potential loss of MAC behavior synchronization between sender and receiver caused by loss of ACK.

C.2 Handling of Multiple Senders or Receivers

In LB-MAC, as the parameter tuning is made between a pair of sender and receiver, a node who serves as a common receiver to multiple senders may decrease ϕ or increase T_r for one sender and then lose the rendezvous with other senders. To address this problem, a receiver records the scheduled T_r and ϕ values with each sender, and chooses the smallest T_r as its wakeup interval and the largest ϕ as its channel checking period. This way, the rendezvous with all senders can be guaranteed.

LB-MAC can also work in mesh-topology networks where each node may have multiple receivers. In this case, a sender node simply transmits each data packet according to the target receiver's parameter setting, so that the rendezvous with the target receiver can be guaranteed.

C.3 Handling of Channel Contention

Under the circumstances where the receiver has a shorter expected lifetime than all its senders, it will keep decreasing the ϕ value. However, when ϕ becomes too small, data packets will be transmitted frequently every $T_s = \phi$ time, which may cause severe contention to the channel and a large number of packet collisions. As a result, senders may waste lots of energy contending for the channel.

To deal with this situation, LB-MAC maintains an online parameter $\phi_{\min}$ as an indicator of the severity of the channel contention. A larger $\phi_{\min}$ corresponds to more severe channel contention. As shown at the top of Fig. 4, $\phi_{\min}$ is doubled/halved when the receiver senses the channel busy/idle after it sends a beacon. The minimum value for $\phi_{\min}$ is set to 10 ms. Then, when the intended new ϕ value is smaller than $\phi_{\min}$, the receiver will notify the sender to set T_s and ρ to ∞. This way, the sender will listen idly for the receiver's beacon to start a data transmission, instead of attempting a data transmission every $T_s = \phi$ time; hence, channel contention can be reduced and energy can be saved at both sender and receiver.

C.4 Handling of Route Changes

In practice, a sender may switch to a new receiver due to route updates. Then, the original receiver may waste energy on unnecessarily long channel checking period or frequent wakeup if it keeps using the ϕ or T_r value scheduled for the stale sender. In LB-MAC, each receiver periodically checks and drops stale senders and the corresponding ϕ and T_r values. Similarly, a sender also drops stale receivers periodically if they don't interact with each other after a certain period. When a sender switches to a new receiver, it waits idly for the new receiver's beacon to establish the initial rendezvous.

V. LB-MAC IMPLEMENTATION

We have implemented LB-MAC in TinyOS 2.1.0 [32]. Fig. 7 shows its composition within the UPMA framework [37], [38], where shaded parts are the main components of LB-MAC:

- *LBMAC scheduler* is the core scheduling component. It resides atop the radio core layer and handles all operations of message processing and parameter tuning, based on the flow charts shown in Figs. 4 and 5.
- *LBMAC adaption code* of the radio core layer provides a variety of low-level supports for the LBMAC scheduler component. Particularly, it monitors channel after sending each beacon and estimates channel contention status based on it.

In the following, we first present the message formats used in LB-MAC and then discuss some implementation issues.

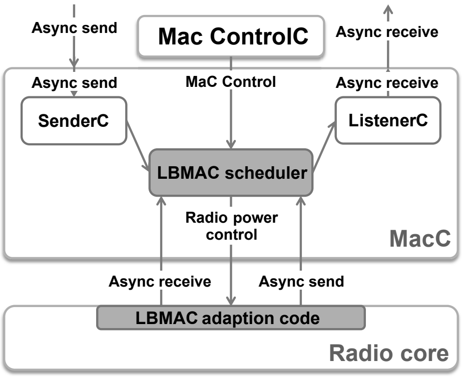


Fig. 7. LB-MAC architecture.

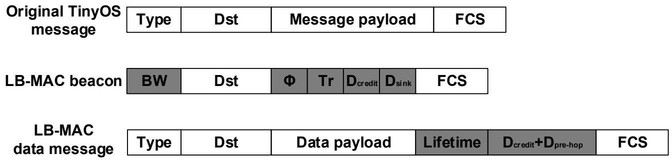


Fig. 8. Message formats used in LB-MAC (shaded fields are added/modified in LB-MAC).

A. Message Formats

Fig. 8 shows the message formats used in LB-MAC, where the shaded fields are the ones added/modified for LB-MAC.

- The beacon message is used by a receiver either as a notification sent upon its wakeup or as a software ACK to acknowledge the reception of a data packet.
- Similar to RI-MAC, LB-MAC reuses the *type* field in the beacon message to carry the backoff window size that will be used by the sender to select its backoff value. Different from RI-MAC and A-MAC, LB-MAC adds four 6-byte fields to each beacon message to carry ϕ, T_r, D_{credit}, and D_{sink} values.
- The sender piggybacks the following information in each data packet: the estimated nodal lifetime, the communication delay of the previous hop, and the tuning credit. This information will be used by the receiver to tune the MAC-layer parameters, as discussed in Section IV-A.

B. Residual Energy Estimation

In order for sensor nodes to make proper decisions on tuning their MAC-layer parameters, it is critical that they can measure/estimate the nodal residual energy and the nodal lifetime. We have designed and fabricated a TelosB power meter kit as shown in Fig. 9 for this purpose. This kit measures the nodal power consumption rate, based on which a node can calculate the total energy consumed so far. The nodal residual energy is the difference between the battery energy capacity [39] and the consumed energy.

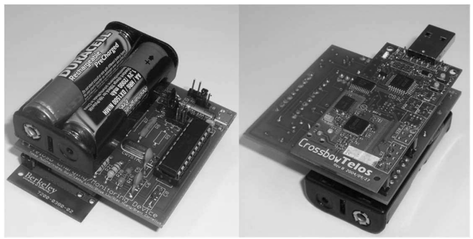


Fig. 9. TelosB power meter kit used in LB-MAC. The working power consumption of this kit is 2.4 μW which is small compared to radio power consumption.

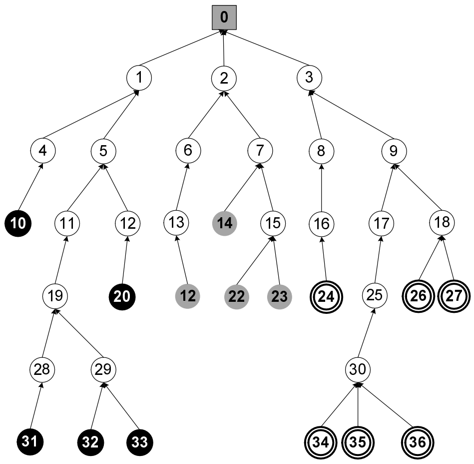


Fig. 10. The initial network topology of the testbed determined by CTP. Sensors in black circles, gray circles and double circles form three different sensing areas.

VI. PERFORMANCE EVALUATION

Experiments have been conducted to evaluate the performance of LB-MAC against X-MAC, RI-MAC, and SEESAW, in terms of *network lifetime*, *data delivery ratio*, *average nodal power consumption*, and *end-to-end data delivery delay*.

A. Experiment Setup

In the experiments, the testbed is composed of 37 TelosB motes, in which node 0 is connected to a computer, and its radio is kept on all the time to serve as the sink. Collection tree protocol (CTP) [40] is used to find the routes for data packet forwarding, and the network topology may vary over time in the experiments; the initial topology established by the routing protocol is shown in Fig. 10. The end-to-end delay requirement D_{e2e} is 6 s in all experiments.

For X-MAC, and RI-MAC, ϕ, ρ, and T_s are set according to Table 2, which reflects the settings in [7] and [8]. The value of T_r for X-MAC and RI-MAC are selected based on empirical results to achieve better network lifetime performances without violating the end-to-end delay requirement. Particularly, in each experiment, X-MAC and RI-MAC are evaluated with T_r set at 0.4, 0.6, 0.8 and 1 s, and the measurements associated with the best network lifetime performances are plotted for X-MAC and RI-MAC in the following figures.

For SEESAW, the initial value of ϕ is set to 30 ms and T_s is set to $\phi/1.2 = 25$ ms [11]. To be comparable with SEESAW, the initial values of both ϕ and T_s in LB-MAC are set to 30 ms. For both SEESAW and LB-MAC, the initial T_r is calculated as $T_r = \frac{D_{e2e}}{\texttt{network diameter}}$, which is based on the end-to-end delay requirement and network diameter. As D_{e2e} is 6 s, and the monitored maximum network diameter is 6, the initial T_r value for both SEESAW and LB-MAC is 1 s.

In the following sections, experiment results are plotted with a 95% confidence interval, except snapshots and traces.

A.1 Lifetime Measurement

During the experiments, we notice that it may take weeks to completely drain a fully-charged battery of sensor nodes. In order to complete all the experiments within a reasonable amount of time while demonstrating the features and performances of evaluated protocols, we study how fast a sensor node consumes a designated small amount of energy, and evaluate its nodal lifetime as the time period during which this designated amount of energy is consumed.[3] This also allows us to start the experiments with nodes at different initial energy levels, which simplifies and speeds up the evaluation process significantly.

B. Static Network Settings

We first compare LB-MAC with other protocols under the scenario of static network settings, in which the sensing event detection pattern, network topology, and packet loss ratio are all fixed. Particularly, the setup is as follows:

- *Static routing paths:* The network topology is set up by CTP at the beginning of experiments and not changed thereafter (by disabling routing updates in CTP).
- *Static sensing events:* Sensing events are assumed to be detected by sensors 24, 26, 27, 34, 35, and 36 only. These sensors (i.e., source nodes) generate data packets at a certain fixed rate and forward them hop by hop to the sink.
- *Static packet loss ratio:* The channel is under the regular lab condition and node software will not drop any packets on purpose; as we measured, the packet loss ratio is negligible in such a lab environment.

B.1 Uniform Initial Nodal Energy

We evaluate LB-MAC under *uniform initial nodal energy*, where all sensor nodes have the same initial nodal energy of 400 Joules. Results are plotted in Figs. 11 and 12.

As shown in Fig. 11(a), LB-MAC yields a longer network lifetime than RI-MAC, X-MAC, and SEESAW under various data generation intervals. When the data generation interval is 2.5 s, LB-MAC extends the network lifetime by about 60% more than RI-MAC and X-MAC, and 30% more than SEESAW. When the data generation interval is 20 s (very low traffic in the network), the improvement of the network lifetime is about 100% over RI-MAC and X-MAC. This is due mainly to the following reasons. As RI-MAC and X-MAC fix the MAC-layer operational parameters, bottleneck nodes (such as node 9) have the heaviest workloads and consume more energy than others; thus, they yield a shorter nodal lifetime, which constrains the network lifetime as shown in Fig. 12(a).

[3]Based on the ratio between the full nodal energy capacity and this designated amount of nodal energy, the measured nodal lifetime can be scaled up to obtain the actual nodal lifetime. Specifically, if the full nodal energy is E_c, the designated nodal energy is E_m and the measured nodal lifetime using the designated energy is ℓ, the actual nodal lifetime can be calculated as $L = \ell \frac{E_c}{E_m}$.

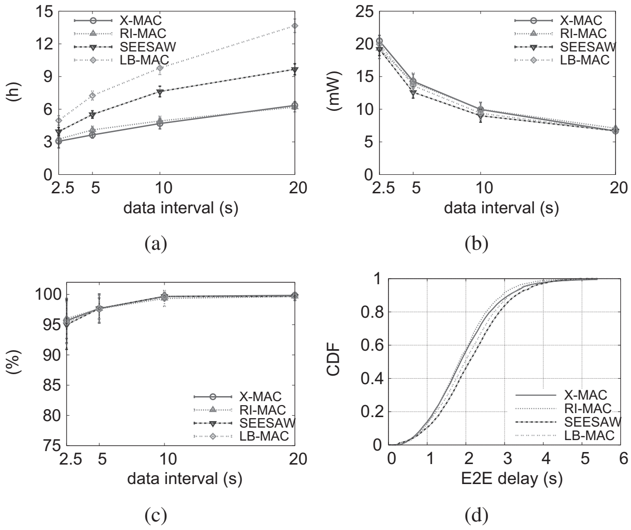


Fig. 11. Performance comparison with uniform initial nodal energy. At data intervals 2.5 s, 5 s, 10 s, and 20 s, the best network lifetime performance for X-MAC is achieved under T_r values of 0.6, 0.8, 1, and 1 s, respectively; the best network lifetime for RI-MAC is obtained under T_r values of 0.8, 0.8, 1, and 1 s, respectively: (a) Network lifetime, (b) average nodal power consumption, (c) data delivery ratio, and (d) CDF of end-to-end delay.

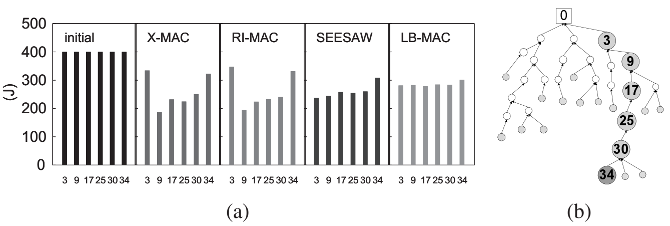


Fig. 12. Snapshots of residual energy with uniform initial nodal energy: (a) Residual energy of nodes 3, 9, 17, 25, 30, and 34 after 2 h of network operation. Data generation interval at source nodes is 5 s and (b) the studied route from node 34 to the sink node 0.

In comparison, LB-MAC dynamically adjusts the MAC-layer parameters to shift communication overhead away from the bottleneck nodes, thus increasing the network lifetime significantly. SEESAW also attempts to balance nodal lifetime by adjusting some of the MAC-layer parameters. However, the parameter adjustment in SEESAW is less effective than that in LB-MAC because SEESAW simply adopts a set of fixed policies that are not adaptive to changes in network conditions. Besides, SEESAW always relies on senders to initiate communications and its performance is degraded in the presence of channel contention; in contrast, when the channel contention is high, LB-MAC switches from sender-initiated to receiver-initiated rendezvous so that channel contention can be alleviated and more energy can be saved.

Fig. 11(b) demonstrates that the longer network lifetime yielded by LB-MAC is achieved without increasing the overall energy consumption in the network. Indeed, LB-MAC maintains similar average nodal power consumption as RI-MAC, X-MAC, and SEESAW. Figs. 11(c) and (d) show that LB-MAC

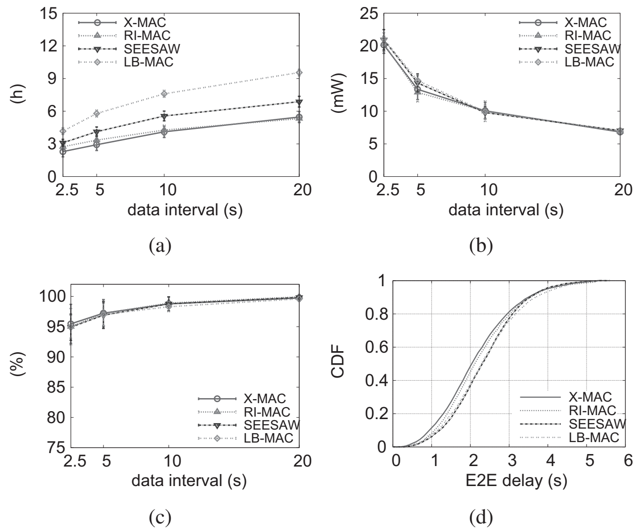


Fig. 13. Performance comparison with non-uniform initial nodal energy. At data intervals 2.5, 5, 10, and 20 s, the best network lifetime performances for X-MAC are achieved under T_r values of 0.6, 0.8, 1, and 1 s, respectively; the best network lifetime performances for RI-MAC are obtained under T_r values of 0.8, 1, 1, and 1 s, respectively: (a) Network lifetime, (b) average nodal power consumption, (c) data delivery ratio, and (d) CDF of end-to-end delay.

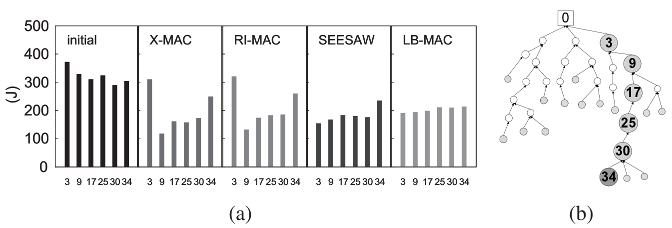


Fig. 14. Snapshots of residual energy with non-uniform initial nodal energy: (a) Residual energy of nodes 3, 9, 17, 25, 30, and 34 after 2 h of network operation. Data generation interval at source nodes is 5 s (b) The studied route from node 34 to the sink node 0.

satisfies the end-to-end delay requirement and achieves a high data delivery ratio.

B.2 Non-Uniform Initial Nodal Energy

As the initial nodal energy may be different in practice, we also evaluate LB-MAC under *non-uniform initial nodal energy*. In this case, the designated amount of energy available at each sensor node varies between 200 and 400 Joules. Results plotted in Fig. 13 show that LB-MAC is able to balance the energy consumption effectively and yield a longer network lifetime.

B.3 A Trace Study

To further illustrate how LB-MAC adaptively tunes the MAC-layer operational parameters to balance the nodal lifetime between neighboring sensor nodes, we examine the experiment that we used to plot the residual energy snapshots in Fig. 14 in more detail, and plot in Fig. 15 the changing traces of the operational parameters of the nodes along the path $34 \to 30 \to 25$: T_s, T_r, and ϕ of node 30, and ϕ of node 25.

We have the following observations:

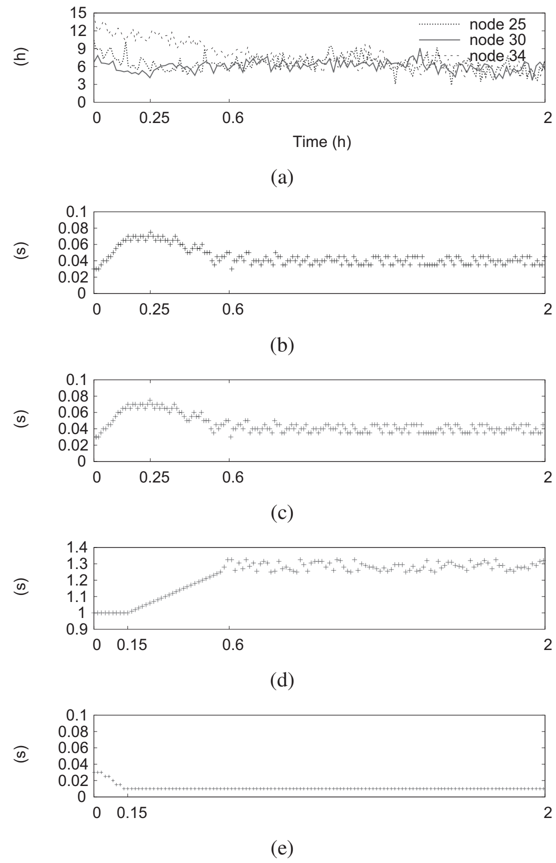


Fig. 15. Changing traces of T_s, T_r, and ϕ of node 30, and ϕ of node 25 along the path $34 \to 30 \to 25$: (a) Comparison of nodal lifetime, (b) ϕ of node 25, (c) T_s of node 30, (d) T_r of node 30, and (e) ϕ of node 30.

- During the time period [0, 0.25 h], as shown in Fig. 15(a), node 30 has a shorter lifetime than both nodes 25 and 34. To balance the nodal lifetime between them, node 25 increases its ϕ to shift communication overhead from node 30 to itself. Correspondingly, node 30 increases its T_s to save energy on transmission and maintain the rendezvous condition. Meanwhile, node 30 also attempts to shift communication overhead to node 34 by first decreasing its ϕ and then increasing its T_r.
- At the time instance of 0.25 h, nodes 25 and 30 have reached a similar nodal lifetime. However, as node 30 still has a shorter lifetime than node 34, it continues to shift communication overhead to node 34. As a result, its lifetime continues to increase, resulting in a lifetime imbalance between itself and node 25. This is the reason why node 25 gradually decreases its ϕ during the time period [0.25 h, 0.6 h].
- Finally, during the time period [0.6 h, 2 h], as all three nodes have a similar nodal lifetime, both ϕ of node 25 and T_s of node 30 stabilize (to fluctuate within a small range around 20 ms) to maintain the lifetime balance between them.

C. Dynamic Network Settings

We also evaluate LB-MAC under more dynamic and time-varying conditions. Specifically, the dynamic network environment settings are as follows:

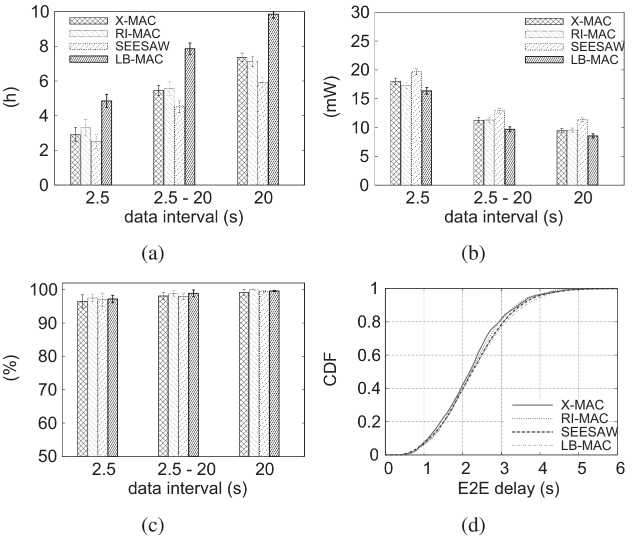


Fig. 16. Performance comparison with non-uniform initial nodal energy and dynamic sensing events. Data interval "2.5–20" means that data packets are generated at an interval uniformly distributed in [2.5 s, 20 s]. Data intervals "2.5" and "20" mean that data packets are generated at an interval uniformly distributed with a mean of 2.5 s and 20 s, respectively; and the deviations are 0.25 s and 2 s, respectively: (a) Network lifetime, (b) average nodal power consumption, (c) data delivery ratio, and (d) CDF of end-to-end delay.

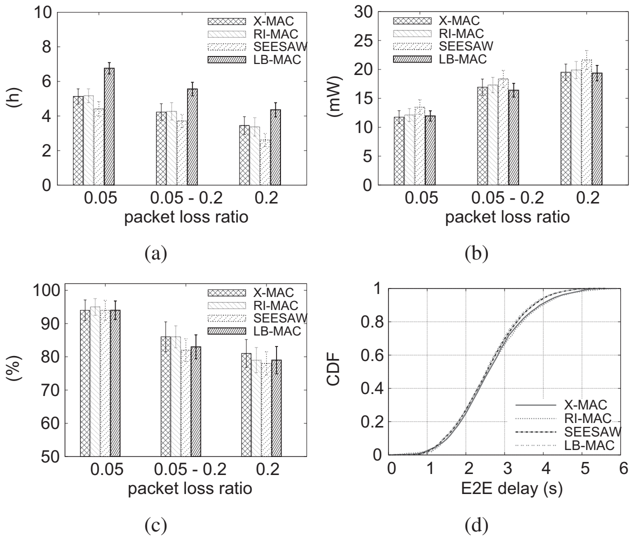


Fig. 17. Performance comparison with non-uniform initial nodal energy and dynamic packet loss ratios. Packet loss ratio interval "0.05–0.2" means that packets are dropped at a ratio uniformly distributed in [0.05, 0.2]. Packet loss ratios "0.05" and "0.2" mean that packets are dropped at ratios uniformly distributed with a mean of 0.05 and 0.2, respectively; and the deviations are 0.005 and 0.02, respectively: (a) Network lifetime, (b) average nodal power consumption, (c) data delivery ratio, and (d) CDF of end-to-end delay.

- *Dynamic routing paths*: The network topology is maintained by CTP protocol, and the topology may vary as experiments continue.
- *Dynamic sensing events:* Sensing events are assumed to be detected by sensors in one of three sensing areas as illustrated in Fig. 10. Every certain period, a sensing area will be active and a sensor in that area will generate data packets and forward them hop by hop to the sink.
- *Dynamic packet loss ratios:* The node software will randomly drop data packets at a certain ratio; this way, we emulate the effect of time-varying packet loss ratios caused by different channel conditions.

C.1 Time-Varying Data Generation Rates

Fig. 16 shows the comparison results when the data generation rate changes over time. In this scenario, LB-MAC also produces a significantly longer network lifetime than the state-of-the-art MAC protocols while maintaining similar end-to-end packet delivery delay, delivery ratio, and average nodal power consumption. The results well demonstrate the robustness and effectiveness of LB-MAC in practical scenarios where (i) the routing paths and traffic patterns are time-varying, and (ii) the data sources are temporally and spatially dynamic. In particular, the superiority of LB-MAC over SEESAW can be observed more clearly from the experiments because SEESAW's fixed and empirical policies (for MAC-layer parameter tuning) do not work well with dynamic events while LB-MAC adapts to network dynamics.

C.2 Time-Varying Packet Loss Ratios

We also evaluate the performance of LB-MAC under a time-varying packet loss ratio by letting each sensor node drop packets with a certain arbitrary ratio; this way, we emulate the changes of communication conditions in a lab environment. The data generation interval is 10 s in these experiments. As shown in Fig. 17(a), when the packet loss ratio is increased, the performance of all evaluated protocols degrades. However, LB-MAC can still yield noticeable lifetime improvement over other protocols.

VII. CONCLUSION AND FUTURE WORK

In this paper, we present a new sensor network MAC protocol, called lifetime-balancing MAC (LB-MAC), which is designed from the perspective of network lifetime maximization. LB-MAC emphasizes collaboration between sensor nodes to benefit the network as a whole, even at the expense of a single node. The key idea is that communicating neighbors adjust their MAC-layer behaviors together in a collaborative manner to shift the communication overhead between them. As a result, nodal lifetime can be balanced between neighbors and network lifetime can be extended. The effectiveness of the proposed scheme is demonstrated via in-depth experimental results.

Future work may be conducted along the following directions. As many schemes have been proposed at layers other than MAC to balance nodal lifetime or energy consumption, we plan to compare LB-MAC with these schemes and study the pros and cons of each approach. Based on the study, we will explore the feasibility and strategy of more advanced balancing techniques through cross-layer integration with middle layer [41], routing layer [42], or services in other network layers [43]. 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 [37], [38], while the advanced scheme is to delegate data transmissions to different nodes [44] 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 can be used when adjusting the MAC-layer parameters.

REFERENCES

[1] G. W. Challen, J. Waterman, and M. Welsh, "IDEA: Integrated distributed energy awareness for wireless sensor networks," in *Proc. ACM MobiSys*, 2010, pp. 35–48.

[2] J.-h. Chang and R. Tassiulas, "Energy conserving routing in wireless Ad-hoc networks," in *Proc. IEEE INFOCOM*, vol. 1, 2000, pp. 22–31.

[3] J. Park and S. Sahni, "An online heuristic for maximum lifetime routing in wireless sensor networks," *IEEE Trans. Comput.*, vol. 55, pp. 1048–1056, Aug. 2006.

[4] W. Wang, V. Srinivasan, and K.-C. Chua, "Using mobile relays to prolong the lifetime of wireless sensor networks," in *Proc. ACM MobiCom*, 2005, pp. 270–283.

[5] K. Lin *et al.*, "Heliomote: Enabling long-lived sensor networks through solar energy harvesting," in *Proc. ACM SenSys*, 2005, pp. 309–309.

[6] X. Jiang, J. Polastre, and D. Culler, "Perpetual environmentally powered sensor networks," in *Proc. ACM IPSN*, 2005, pp. 463–468.

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

[8] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," in *Proc. ACM SenSys*, 2008, pp. 1–14.

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

[10] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis, "Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless," in *Proc. ACM SenSys*, 2010, pp. 1–14.

[11] R. Braynard, A. Silberstein, and C. S. Ellis, "Extending network lifetime using an automatically tuned energy-aware MAC protocol," in *Proc. Springer-Verlag EWSN*, 2006, pp. 244–259.

[12] R. Jurdak, P. Baldi, and C. Videira Lopes, "Adaptive low power listening for wireless sensor networks," *IEEE Trans. Mobile Computing*, vol. 6, pp. 988–1004, Aug. 2007.

[13] C. M. Vigorito, D. Ganesan, and A. G. Barto, "Adaptive control of duty cycling in energy-harvesting wireless sensor networks," in *Proc. IEEE SECON*, 2007, pp. 21–30.

[14] A. Meier, M. Woehrle, M. Zimmerling, and L. Thiele, "ZeroCal: Automatic MAC protocol calibration," in *Proc. IEEE DCOSS*, 2010, pp. 31–44.

[15] C. J. Merlin and W. B. Heinzelman, "Duty cycle control for low-power-listening MAC protocols," *IEEE Trans. Mobile Computing*, vol. 9, pp. 1508–1521, November 2010.

[16] Y. Peng, Z. Li, D. Qiao, and W. Zhang, "Delay-bounded MAC with minimal idle listening for sensor networks," in *Proc. IEEE INFOCOM*, 2011, pp. 1314–1322.

[17] Z. Li, M. Li, and Y. Liu, "Towards energy-fairness in asynchronous duty-cycling sensor networks," in *Proc. IEEE INFOCOM*, 2012, pp. 801–809.

[18] M. Zimmerling, F. Ferrari, L. Mottola, T. Voigt, and L. Thiele, "pTunes: Runtime parameter adaptation for low-power MAC protocols," in *Proc. ACM IPSN*, 2012, pp. 173–184.

[19] M. T. Hansen, B. Kusy, R. Jurdak, and K. Langendoen, "AutoSync: Automatic duty-cycle control for synchronous low-power listening," in *Proc. IEEE SECON*, 2012, pp. 139–147.

[20] A. El-Hoiydi and J.-D. Decotignie, "WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks," in *Proc. ALGOSENSORS*, 2004.

[21] P. Hurni and T. Braun, "MaxMAC: A maximally traffic-adaptive MAC protocol for wireless sensor networks," in *Proc. EWSN*, 2010.

[22] M. Sha, G. Hackmann, and C. Lu, "Energy-efficient low power listening for wireless sensor networks in noisy environments," in *Proc. ACM IPSN*, 2013, pp. 277–288.

[23] Q. Li, J. Aslam, and D. Rus, "Online power-aware routing in wireless Ad-hoc networks," in *Proc. ACM MobiCom*, 2001.

[24] L. Lin, N. B. Shroff, and R. Srikant, "Asymptotically optimal power-aware routing for multihop wireless networks with renewable energy sources," *IEEE/ACM Trans. Netw.*, 2007.

[25] Y. Gu and T. He, "Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links," in *Proc. ACM SenSys*, 2007, pp. 321–334.

[26] Y. Gu and T. He, "Dynamic switching-based data forwarding for low-duty-cycle wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 10, pp. 1741–1754, 2011.

[27] S. Cui, R. Madan, A. Goldsmith, and S. Lall, "Joint routing, MAC, and link layer optimization in sensor networks with energy constraints," in *Proc. IEEE ICC*, 2005, pp. 725–729.

[28] R. Madan, S. Cui, S. Lall, and N. Goldsmith, "Cross-layer design for lifetime maximization in interference-limited wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 5, no. 10, pp. 3142–3152, 2006.

[29] H. Wang, X. Zhang, F. Nait-Abdesselam, and A. Khokhar, "Cross-layer optimized MAC to support multihop QoS routing for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 5, pp. 2556–2563, 2010.

[30] S. Ehsan, B. Hamdaoui, and M. Guizani, "Cross-layer aware routing approaches for lifetime maximization in rate-constrained wireless sensor networks," in *Proc. IEEE GLOBECOM*, 2010, pp. 1–5.

[31] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *Proc. IEEE INFOCOM*, 2006, pp. 1–13.

[32] TinyOS Community, "TinyOS." [Online]. Available: http://www.tinyos.net/

[33] Texas Instruments, "CC2420 datasheet." [Online]. Available: http://www.ti.com/lit/ds/symlink/cc2420.pdf

[34] Willow Technologies, "TelosB mote datasheet." [Online]. Available: http://www.willow.co.uk/TelosB_Datasheet.pdf

[35] S. Guo, Y. Gu, B. Jiang, and T. He, "Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links," in *Proc. ACM MobiCom*, 2009, pp. 133–144.

[36] X. Wang, X. Wang, G. Xing, and Y. Yao, "Dynamic duty cycle control for end-to-end delay guarantees in wireless sensor networks," in *Proc. IEEE IWQoS*, 2010, pp. 1–9.

[37] K. Klues, G. Hackmann, O. Chipara, and C. Lu, "A component based architecture for power-efficient media access control in wireless sensor networks," in *Proc. ACM SenSys*, 2007, pp. 59–72.

[38] UPMA - Washington University St. Louis, "UPMA package: Unified power management architecture for wireless sensor networks." http://tinos.cvs.sourceforge.net/tinyos/tinyos-2.x/contrib/wustl/upma/

[39] "All about batteries." [Online]. Available: http://www.allaboutbatteries.com /Energy-tables.html

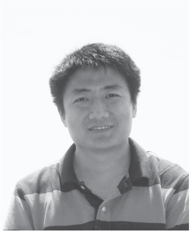
[40] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proc. ACM SenSys*, 2009, pp. 1–14.

[41] Z. Li, Y. Peng, D. Qiao, and W. Zhang, "LBA: Lifetime balanced data aggregation in low duty cycle sensor networks," in *Proc. IEEE INFOCOM*, 2012, pp. 1844–1852.

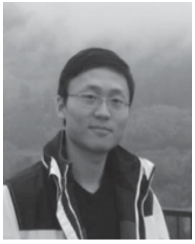
[42] Z. Li, Y. Peng, W. Zhang, and D. Qiao, "J-RoC: A joint routing and charging scheme to prolong sensor network lifetime," in *Proc. IEEE ICNP*, 2011, pp. 373–382.

[43] K.-W. Fan, Z. Zheng, and P. Sinha, "Steady and fair rate allocation for rechargeable sensors in perpetual sensor networks," in *Proc. ACM SenSys*, 2008, pp. 239–252.

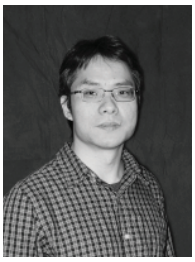
[44] Y. Sun, O. Gurewitz, S. Du, L. Tang, and D. B. Johnson, "ADB: An efficient multihop broadcast protocol based on asynchronous duty-cycling in wireless sensor networks," in *Proc. ACM SenSys*, 2009, pp. 43–56.



Yang Peng received his Ph.D. degree in Computer Science from Iowa State University in 2014. He is currently an Assistant Professor in the Division of Computing and Software Systems, University of Washington Bothell, Bothell, WA, USA. His research interests include Internet of Things, Wireless Sensor Networks, and Cyber-Physical Systems.



Zi Li received his Ph.D. degree in Computer Science from Iowa State University in 2013. He is currently a Staff Software Engineer in LinkedIn, Mountain View, CA, USA. His research interests include Social Networks, Wireless Sensor Networks, and Mobile Computing.



Wensheng Zhang got his Ph.D. degree in Computer Science and Engineering from The Pennsylvania State University. He joined the Department of Computer Science at Iowa State University in 2005, where he is now an Associate Professor. His research interests include distributed systems, networks, and security.

Daji Qiao received the Ph.D. degree in Electrical Engineering: Systems from the University of Michigan, Ann Arbor in 2004. He is currently an Associate Professor in the Department of Electrical and Computer Engineering, Iowa State University, Ames, Iowa, USA. His research interests include Sensor Networks and Internet of Things, Wireless Networking and Mobile Computing, and Cyber Security. He is a member of the IEEE and ACM.