

## Regular paper

A prediction-based asynchronous MAC protocol for heavy traffic load in wireless sensor networks<sup>☆</sup>Chuchu Dong, Fengqi Yu<sup>\*</sup>

Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China  
University of Chinese Academy of Sciences, Beijing, China

## ARTICLE INFO

## Article history:

Received 20 July 2017

Accepted 7 September 2017

## Keywords:

Wireless sensor network  
Medium access control  
Asynchronous duty cycle  
Energy efficient

## ABSTRACT

Radio transceiver often consumes most of energy in a sensor node. To achieve low power consumption, every node periodically schedules its radio transceiver into sleep or active state, which is usually called duty cycle mechanism in MAC protocol. In this paper, we design a novel Prediction-Based Asynchronous MAC protocol (PBA-MAC) for heavy traffic load in wireless sensor networks. PBA-MAC applies an efficient wakeup mechanism to save the energy of a sensor node as much as possible. It reduces communication cost by enabling a sender to predict the intended receiver's wakeup time, without introducing unnecessary idle listening or overhearing cost. In addition, it introduces an exponential advance mechanism to handle the prediction error caused by clock drift and hardware platform latency. More importantly, its backcast-based retransmission characteristic can efficiently resolve severe wireless collisions caused by concurrent traffic flows. The PBA-MAC is implemented in Contiki operating system. We compare PBA-MAC to some existing energy-efficient MAC protocols. The experimental results show that PBA-MAC significantly improves network energy efficiency and reduces packet delivery latency even under heavy traffic load.

© 2017 Elsevier GmbH. All rights reserved.

## 1. Introduction

Wireless sensor networks (WSNs) provide distributed and self-organized solution for many industrial applications. Typically, a WSN consists of a large number of low-cost sensor nodes that are usually battery-powered. Therefore, in order to prolong the network lifetime, sensor nodes must reduce energy consumption as much as possible [1–4]. In a wireless sensor node, its radio transceiver often consumes the most energy when actively transmitting packets, as well as passively listening or receiving packets from neighbors. Thus, the sensor nodes usually turn off their radio as long as possible to save power. Since neighbor nodes should turn on radio to communicate with each other, they use the duty cycle mechanism to periodically schedule radio transceiver between sleep and active state [5–8]. The average duty cycle is defined as the fraction of the time nodes are awake during their lifetime. Over the years, numerous duty cycle mechanisms have

been proposed in MAC protocols and they are either synchronous or asynchronous [9–13].

Synchronous duty cycle mechanism synchronizes neighbor nodes' sleep and wakeup time to save energy, but multihop time synchronization introduces high overhead [14,15]. In contrast, asynchronous MAC protocols do not depend on any prior synchronization and maintain independent wakeup schedule. Under light traffic load, most of existing asynchronous MAC protocols are energy efficient. However, their energy efficiency decreases greatly when there are multiple concurrent traffic flows, so they cannot scale well in large and dense networks. The simultaneous packet sending of multiple nodes cause severe wireless collisions under heavy traffic load, which results in unnecessary energy consumption and large packet delay [16,17].

This paper describes a novel Prediction-Based Asynchronous MAC protocol (PBA-MAC). PBA-MAC is designed to be easy to understand and implement. PBA-MAC introduces a more energy efficient wakeup mechanism than previous ones. In PBA-MAC, neighbor nodes can precisely predict each other's wakeup time and reliably complete packet transmission without introducing unnecessary idle listening or overhearing cost. An efficient backcast-based collision resolving mechanism has been proposed in PBA-MAC to retransmit packets under heavy traffic load. Our

<sup>☆</sup> The authors are with Shenzhen Institutes of Advanced Technology, Chinese Academy of Science.

<sup>\*</sup> Corresponding author at: Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China.

E-mail address: [fq.yu@siat.ac.cn](mailto:fq.yu@siat.ac.cn) (F. Yu).

PBA-MAC is implemented in Contiki on real sensor nodes. We compare PBA-MAC with some existing asynchronous MAC protocols (i.e., ContikiMAC, WiseMAC and PW-MAC) under different scenarios. The experimental results show that PBA-MAC achieves the highest performance among all protocols.

This paper is organized as follows. In Section 2, we introduce the related work on asynchronous MAC protocols for WSN. Section 3 presents the details of PBA-MAC design. In Section 3.4, we evaluate the performance of PBA-MAC and present the experimental results. We conclude this paper in Section 5.

## 2. Related work

An efficient MAC protocol can significantly reduce energy consumption of sensor nodes and greatly extend network lifetime [18–20]. There are many MAC protocols for wireless sensor networks that have been proposed and implemented [21,5,22]. All of them are focusing on the mechanism of duty cycle [23–25]. In this section, we provide an overview of WSN MAC protocols with more recently proposed work.

Duty cycle mechanisms of WSN MAC protocols can be either synchronous or asynchronous. Many synchronous MAC protocols such as RMAC [26] and DW-MAC [27] synchronize neighbor nodes' active and sleep time to achieve energy efficiency. However, the strict time synchronization in these protocols introduces additional power consumption in multihop networks [28,29]. In addition, the performance of synchronous MAC protocols will decrease with network size and traffic loads increasing, since the fixed sleep scheduling will cause severe channel collision. In contrast, asynchronous MAC protocols have the advantage of not requiring synchronization and allow nodes to wake up asynchronously and independently base on their own schedules, which leads to less wireless contention and better channel utilization. With independent wakeup schedules, asynchronous MAC protocols can greatly reduce energy consumption caused by idle listening and overhearing, but they must design a reliable and efficient mechanism for the packet transmission between neighbor nodes.

The research community has explored many different variants of asynchronous protocols, which can be divided into two main categories: sender-initiated and receiver-initiated. A sender in sender-initiated MAC protocols will transmit a preamble before sending data packet. The preamble transmission should last longer than the intended receiver's maximum wakeup interval. This limitation ensures that the preamble can be listened by all intended receivers. Once a node does not detect available preamble transmission when conducts CCA (Clear Channel Assessment) check, it can go to sleep immediately. This method can eliminate unnecessary idle listening time of the intended receiver when there are few packets sent for it under light traffic load. Although the preamble transmission can achieve low duty cycle, it will significantly reduce channel utilization and limit network throughput by interfering with the data packet transmission of neighboring nodes. For this reason, receiver-initiated MAC protocols attempts to reserve more channel resources for data packet transmission. In recent years, many schemes have been proposed to transmit data packet reliably with minimizing the length of preamble.

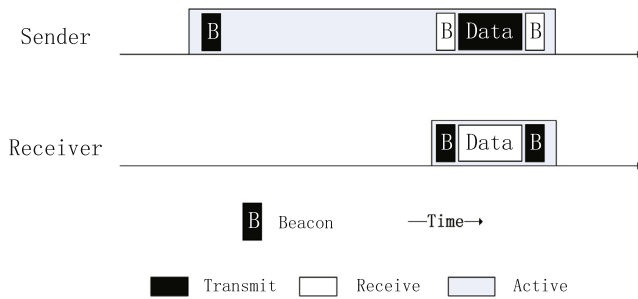
For easy implementation, the researchers first designed an asynchronous communication model called Low Power Listening (LPL), which was initially proposed on the Mica wireless platform [30]. Based on LPL mechanism, J. Polastre et al. proposed B-MAC [31] in which all nodes wake up independently base on their own schedules. B-MAC uses a long wakeup signal as a preamble prior to data packet transmission. X-MAC [32] adopts a strobe preamble which is divided into a sequence of short preambles. For better performance, Klues Kevin et al. improved the short

preambles by using the pending data packet itself in a variation of X-MAC in UPMA architecture [33]. ENBMAC [34] uses the same method but includes the timing information on when the data packet transmission will begin in each short preamble. By this way, once the target receiver in ENBMAC receives a complete short preamble and obtains the time information of the sender, it can go to sleep immediately and compute the next wakeup time for receiving data.

Many protocols share the similarity to divide a long preamble into short packets, such as SpeckMAC [35], SyncWUF [36], TICER [37], CMAC [38] and MX-MAC [39]. Although these protocols include different information in the short preamble for different requirements, none of them reserves a gap between two short packets. For more improvement in performance, some researchers prefer to insert a gap between two short packets in the long preamble to allow target receiver to send an acknowledgement. Under light traffic load, this early acknowledgement can greatly reduce the preamble length and efficiently avoid the overhearing problem. However, this design introduces more energy consumption in idle listening because the duration of conducting channel assessment must be larger than the gap between two short packets. Some protocols [35,39] use data packet itself as the short preamble, which makes the channel assessment last even longer and increases the idle listening period.

The protocols mentioned above reduce the preamble length by dividing a long preamble into chunks and inserting some useful information into these chunks. However, the wireless channel need to be occupies by preamble transmission even when the intended receiver stays in sleep state, resulting in unnecessary energy consumption and low channel utilization. Furthermore, energy is greatly wasted by overhearing the preamble transmission at non-target nodes. Another more energy efficient way to reduce preamble length is wakeup schedule learning. WiseMAC [40] and ContikiMAC [41,42] are the representative protocols using this method. In these protocols, neighboring nodes are able to learn the sleep and wakeup schedules of each other. To efficiently enable this learning, after receiving a data packet, the receiver replies with an acknowledgement message and also embeds its next wakeup time in this message. By decoding the wakeup information and considering the clock drift error, a sender can deduces the receiver's next wakeup time and begins the preamble transmission to this receiver just before then. This method can significantly reduce the sender's preamble length and greatly help to save energy and improve channel utilization. Although schedule learning enables better energy efficiency, maintaining and updating the wakeup schedules of all neighboring nodes introduces additional overhead, which motivates researchers to propose some receiver-initiated MAC protocols [43–45].

The basic idea of receiver-initiated MAC protocols is shifting communication initiation from the sender side to the receiver side. There are many receiver-initiated MAC protocols have been proposed for general sensor networks, such as AS-MAC [46], RI-MAC [45], A-MAC [47] and L-MAC [48]. To squeeze more room for data packet transmission, these protocols adopts the active low power probing (LPP) rather than the passive low power listening (LPL). In low power probing (LPP) mechanism, as shown in Fig. 1, a node periodically wakes up and broadcasts a beacon. Once receiving a wakeup beacon from the intended receiver, the sender concludes that it can transmit data packet to the receiver now. In this way, the wireless channel is not occupied by the sender's preamble transmission and can be used by other nodes for data transmission before the target receiver wakes up. PW-MAC [49] further improves LPP-based protocols by enabling senders to predict intended receiver's wakeup beacon. This mechanism eliminates the sender's energy consumption caused by idle listening for wakeup beacon. A-MAC applies a more efficient wakeup mecha-



**Fig. 1.** Low power probing (LPP) mechanism.

nism to avoid unnecessary idle listening and improve transmission efficiency. However, the performance of LPP-based protocols will significantly decrease under heavy traffic load. None of these protocols has proposed an efficient method to solve severe channel collision caused by concurrent traffic flows, which results in more energy consumption and larger packet delays.

There are some protocols use TDMA to improve channel utilization, such as Z-MAC [50], TreeMAC [51], O-MAC [52], OPWUM [53] and VeMAC [54]. These protocols divide time into frames. Each frame is divided into  $n$  slots. To achieve more efficient reuse of time slots, some slots can be reassigned for data receiving rather than sending when a node has no data to transmit. This adaptive mechanism leads to more energy efficient with introducing less overhead and avoiding unnecessary idle listening. However, these TDMA MAC protocols have the same challenges as other asynchronous ones, which include severe channel collision resolving and optimal packet delay guarantee. The major difference between our PBA-MAC and these MAC protocols mentioned above is that PBA-MAC can efficiently avoid persistent channel collision even under heavy traffic load. As a result, PBA-MAC significantly improves network energy efficiency and reduces packet delivery latency from sensor node to destination.

### 3. Protocol design

In this section, we first introduce an overview of PBA-MAC protocol and then discuss its detail design.

### 3.1. Overview

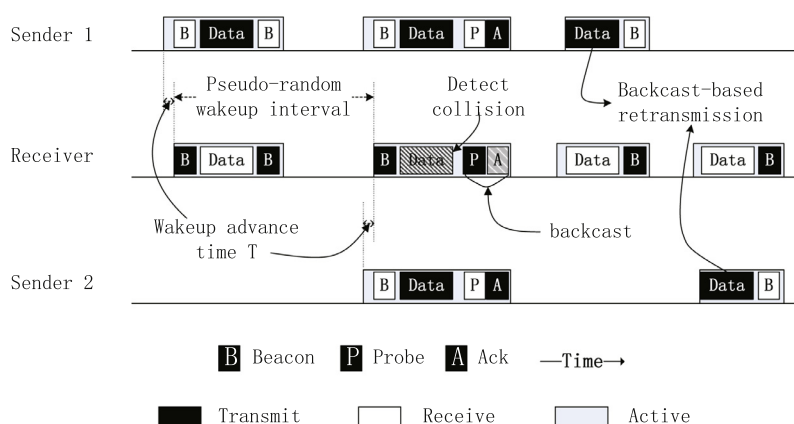
PBA-MAC cannot only reduce energy consumption in light traffic load scenarios where few channel contention is happening, but also achieve energy efficiency when there are multiple concurrent

data transmissions under heavy traffic load. The overview of PBA-MAC protocol is presented in Fig. 2. In PBA-MAC, a node periodically wakes up and broadcasts a beacon to inform neighbor nodes to initiate data transmission. Upon receiving the wakeup beacon, the sender can transmit data to the receiver immediately. Unlike a node stays awake once it has data to send as shown in Fig. 1, PBA-MAC introduces a predictive wakeup mechanism to compute next wakeup time of target receiver so that the sender can wake up just before the receiver does. This prediction-based mechanism can significantly reduce energy consumption of senders and will be used for packet collision resolving.

Under heavy traffic load, multiple nodes may transmit data to the same receiver simultaneously and severe channel collision will happen. This channel collision results in the demand for retransmitting packets. Previously, most of MAC protocols use Binary Exponential Backoff (BEB) strategy [49,55] to resolve collision, as shown in Fig. 3. When the target receiver detects a collision, it replies with an acknowledgement beacon which contains a larger backoff time window. The contending senders increase their backoff time with Binary Exponential Backoff (BEB) strategy to retransmit data packet. The receiver and senders should stay awake until every sender's transmission is succeed, which not only increases duty cycle, but also end-to-end delivery latency. In addition, overhearing a transmission may incorrectly trigger the recovery mechanism, leading to more redundant beacons and then more collisions. To solve this problem, PBA-MAC introduces a backcast-based retransmission mechanism which can avoid persistent collisions. We will describe the details of this mechanism in the following subsection.

### 3.2. Backcast-based retransmission

Backcast is first proposed by Dutta et al. [56]. A backcast is a link-layer frame exchange in which a single radio frame transmission triggers zero or more acknowledgement frames that interfere non-destructively at the initiator. Fig. 4 illustrates a backcast exchange involving three nodes. The two responders have their radio configured to automatically acknowledge any received frames. The backcast exchange begins with the initiator transmitting a probe frame to the hardware broadcast address. Both responders receive the probe and they both transmit identical hardware acknowledgements (HACKs). Although these two HACKs collide at the initiator, this collision is non-destructive, allowing the initiator to successfully decode the HACK frame and conclude that at least one of its neighbors responded. Authors evaluated the performance of backcast on the commercial CC2420 radio. Experimental results confirm that a receiver can successfully



**Fig. 2.** Overview of PBA-MAC.

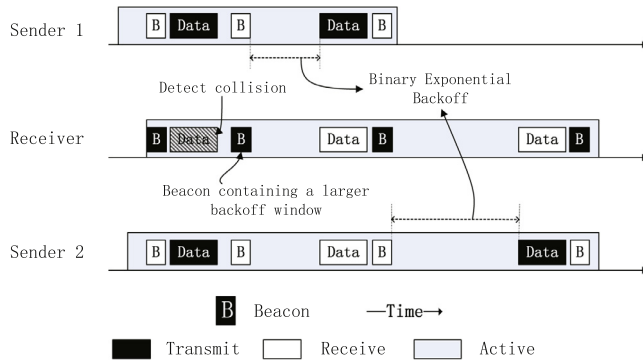


Fig. 3. Binary Exponential Backoff (BEB) strategy.

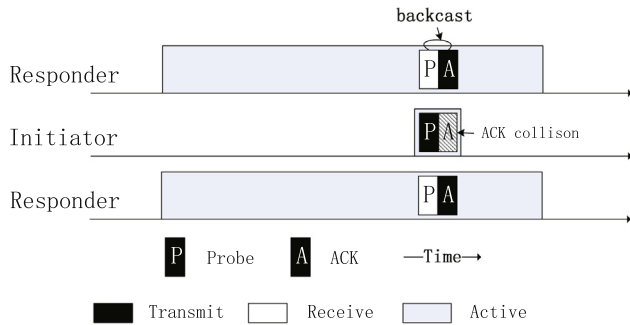


Fig. 4. Backcast.

decode the HACK frame even when twelve identical hardware acknowledgements simultaneously responded to it.

Based on backcast, we introduce an energy efficient retransmission mechanism in PBA-MAC, which uses backcast primitive to achieve single-hop collaborative feedback of contending senders. As shown in Fig. 2, when a collision is detected at the receiver, a backcast exchange begins with the receiver broadcasting a probe frame. Upon receiving this probe frame, all contending senders learn that they should retransmit data packets instead of sending wakeup beacon in their next wakeup time. In addition, they transmit identical acknowledgements to the receiver. Although these concurrent acknowledgements cause collisions, they are non-destructive to be decoded so that the intended receiver knows which senders will retransmit data. After backcast exchanging, both contending senders and the receiver go to sleep immediately. Then, the receiver wakes up to receive data packets by predicting the next wakeup time of every potential sender. Because these concurrent senders retransmit data based on their own wakeup schedule, PBA-MAC can avoid persistent collisions and significantly reduces duty cycle both at senders and receiver. Furthermore, since a node wakes up at least once during the maximum wakeup interval, all contending senders can immediately complete packet retransmission in one wakeup cycle, resulting in less packet delivery latency.

### 3.3. Predictive wakeup mechanism

The goal of PBA-MAC is for a sender to wake up and turn on its radio right before the intended receiver wakes up. To enable senders and receivers to accurately predict the future wakeup time of each other, we require every node on PBA-MAC to compute its wakeup time using a pseudo-random function generator. Applying a pseudo-random wakeup schedule rather than a fixed schedule, we can avoid the possibility of neighboring nodes consistently

waking up at the same time, as such occurrences would significantly increase the chances of collision from senders that are hidden with each other.

There are many ways to build pseudo-random number generators suitable for use in PBA-MAC. For the sake of simplicity, we take the linear congruential generator (LCG) [57,58] in Eq. (1) to generate pseudo-random number. LCG generates a pseudo-random number as follow:

$$X_{n+1} = (aX_n + c) \bmod m \quad (1)$$

where  $m$  ( $m > 0$ ) is the modulus,  $a$  ( $0 < a < m$ ) is the multiplier,  $c$  ( $0 \leq c < m$ ) is the increment, and  $X_n$  ( $0 \leq X_n < m$ ) is the current seed. Each  $X_{n+1}$  generated can be used as a pseudo-random number and become next seed. In our implementation, the parameters  $a$ ,  $c$ , and  $m$  are configured as 20, 7, and 1000. We choose this configuration by following the principles suggested by Knuth [58], such that the LCG will generate all possible numbers in a different pseudo random order each time before any possible number is repeated. Node ID will be used as first seed.

Fig. 2 illustrates the predictive-wakeup mechanism of PBA-MAC. Each node periodically wakes up and broadcasts a beacon, denoted as B in the figure. This beacon announces that I am awake and ready to receive data packets. The interval between wakeup beacons is calculated by a pseudo-random number generator such as the one in Eq. (1).

In PBA-MAC, if a new node  $S$  joins network and has pending data packet, it stays active silently while waiting for the wakeup beacon from the intended receiver  $R$ . After receiving  $R$ 's wakeup beacon,  $S$  then sets a special flag in its data packet header to request  $R$ 's prediction information. Once  $R$  receives this data packet, it sends another beacon that serves as acknowledgement of the data packet reception. In addition,  $R$  also embeds its current time and pseudo-random number generator parameters in the ACK beacon. The current time of  $R$  is used to compute the clock difference between  $S$  and  $R$ . With the prediction information received from the ACK beacon, node  $S$  can easily deduce all future wakeup times of  $R$ . In the future, if  $S$  has another data packet for  $R$ ,  $S$  wakes up shortly before the predicted wakeup time of  $R$ . In this way, PBA-MAC reduces idle listening time to almost 0 once the prediction information of the intended receiver has been learned by the sender, which greatly improves energy efficiency.

### 3.4. Handling prediction error

Clock drift, hardware latency, and the latency of operation system may cause the prediction error. We define the difference between the predicted and actual wakeup time of a node as prediction error. Fig. 5 shows how the prediction error develops over time between two different arbitrarily nodes. For easy comparison, we can assume the two nodes having the same clock drift rate because they are placed in the same environment and use the same operating system. The sender  $S$  requests the prediction information of the receiver  $R$  only at the beginning of the experiment. The average wakeup interval of  $R$  is 1 s and  $S$  continuously predicts wakeup times of  $R$ . Node  $S$  and  $R$  sends every value of predicted and actual wakeup time to PC through UART interface respectively, so that we can easily measure the difference between them. The experiments lasted more than 5000 s and each measurement point in Fig. 5 indicates the prediction error at a particular time. In Fig. 5, we can see that the prediction error between two different nodes will increase over time, varying from 0 to 35 ms.

It is very essential to carefully handle prediction error because the target receiver may actually wake up before or after the predicted time, which is indicated by the positive or negative value of prediction error in Fig. 5. If the target receiver wakes up earlier, the sender will miss the wakeup beacon of the receiver. As a result,



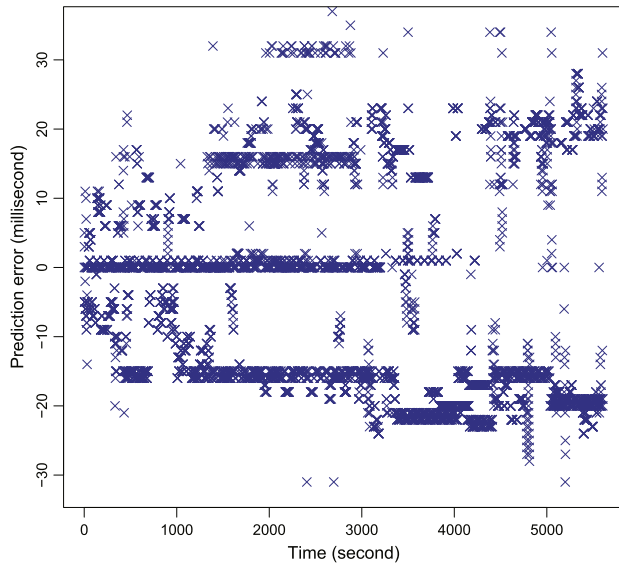


Fig. 5. The prediction error between two different nodes.

the packet transmission will be delayed at least one wakeup interval and the sender's duty cycle will be significantly increased. On the other hand, if the target receiver actually wakes up much later than the predicted time, the sender must stay awake to wait for the receiver's wakeup beacon, introducing more idle listening and also increasing sender's duty cycle. Therefore, to compensate prediction error, a sender in PBA-MAC advances its wakeup by a short time  $T$ . The choice of  $T$  is the tradeoff among packet delay, power consumption, and successful reception of wakeup beacon, which will be discussed in Section 3.4.

In addition, if the prediction error is larger than the wakeup advance time  $T$ , the sender will miss the receiver's wakeup beacon. For maintaining a small duty cycle, the sender stays awake at most twice the wakeup advance time and goes back to sleep. The reason for limiting the sender's waiting time to be  $2T$  is that the predicted wakeup time of the receiver can be earlier or later than its actual wakeup time. After missing a receiver's wakeup beacon, PBA-MAC introduces the exponential advance algorithm to enable the sender to quickly rendezvous with the receiver again.

The sender invokes the exponential advance algorithm by doubling its current wakeup advance time  $T$  for the same receiver in next wakeup cycle. In this way, the sender's waiting time  $2T$  for overhearing the receiver's wakeup beacon is also doubled. The sender repeats this iterative procedure until receiving a wakeup beacon from the receiver. If the sender is still unable to rendezvous with the receiver after the sender's waiting time has expanded to surpass the maximum wakeup interval, the sender gives up sending data to this receiver. This is because the target receiver may have been unreachable otherwise it should wake up at least once during the maximum wakeup interval. Once successfully receiving a wakeup beacon from the receiver, the sender requests this receiver to return its current prediction information in the ACK beacon. Then the sender resets its wakeup advance time  $T$  to its initial value for sending future data packets. Fig. 6 shows an example of the operation of exponential advance algorithm.

### 3.5. Prediction-based wakeup algorithm

In this subsection, we summarize the detailed description of prediction-based wakeup algorithm in PBA-MAC, which is presented in Fig. 7. When a sender  $S$  has data packet sent to the intended receiver  $R$ , it need to request the prediction information

of  $R$ , including the parameters of pseudo-random number generator of  $R$  (randNum[ $R$ ], 2 bytes), the seed (randSeed[ $R$ ], 4 bytes) and the current time of  $R$  (curTime[ $R$ ], 4 bytes). The total additional memory cost due to embedding these parameters in the ACK beacon is only 10 bytes which is acceptable, because the default packet size is 64 bytes. By repeatedly invoking the prediction algorithm,  $S$  can deduce all future wakeup times of  $R$ . If  $S$  finds a next-WakeupTime[ $R$ ] is at least  $T$  larger than the curTime[ $R$ ], it just knows when to wake up for sending packet. The value  $T$  denotes the wakeup advance time of  $S$ , and  $T_{\max\text{Interval}}$  denotes the maximum wakeup interval.

## 4. Performance evaluation

The performance of PBA-MAC will be evaluated in this section. We present the experiment results of two performance metrics: average duty cycle and packet delivery latency. Packet delivery latency measures the average time cost by delivering packets from the source node to destination.

We conducted the experiments on the testbed provided at Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, where 65 nodes are deployed in a student apartment, including 50 sensor nodes, 14 relay nodes and 1 destination node. All of these nodes are built of STM8L15x microcontroller with 4 kbytes of RAM and a Si4438 radio operating at 433 MHz with a maximum data rate of 500 kbps. They communicate wirelessly and are self-organized into an ad hoc network. The network topology and nodes deployed in the testbed are presented in Fig. 8a and b respectively. In the experiments, sensor nodes are installed at each floor to sample the environment temperature and generate one 48 bytes data packet. Each sensor node periodically sends packet to the destination at different intervals varying from 500 to 1500 ms. All relay nodes can forward data packets from different sensor nodes to the destination. To evaluate the impact of wireless collisions on packet transmission, all nodes of the adjacent floor are placed within their radio interference range. Such interfering traffic flows will cause severe channel collisions. PBA-MAC is implemented using C programming language in Contiki operating system. In our experiment, each node computes its wakeup interval by using the pseudo-random generator in Eq. (1), where the parameters  $a$ ,  $c$ , and  $m$  are set to be 20, 7, and 1000. Node ID will be used as first seed. As a result, each node's wakeup interval is randomly distributed between 500 to 1500 ms. Our test results are compared with those of some existing energy-efficient MAC protocols, such as ContikiMAC, WiseMAC and PW-MAC. In the comparison, exactly the same number of data packets is transmitted for all protocols. The final results are obtained by averaging the measurements.

### 4.1. Wakeup advance time

As mentioned in Section 3, a sender in PBA-MAC should advance its wakeup by a short time  $T$  to compensate prediction error caused by clock drift and system-dependent latency. To evaluate the impact of different wakeup advance time  $T$  on real sensor nodes, we do experiments on 10 pairs of sensor nodes. Each pair of sensor nodes comprises a sender and receiver. The average sender duty cycle and packet delivery latency are evaluated under different  $T$ , as shown in Fig. 9a and b.

Obviously, large  $T$  reduces the chance of sender missing the intended receiver's wakeup beacon to prevent transmission failure, but it increases duty cycle since the sender must stay awake for a longer time. On the other hand, the decrease of  $T$  results in reducing wakeup time of sender, but it decreases the chance of overhearing the intended receiver's wakeup beacon. Missing wakeup

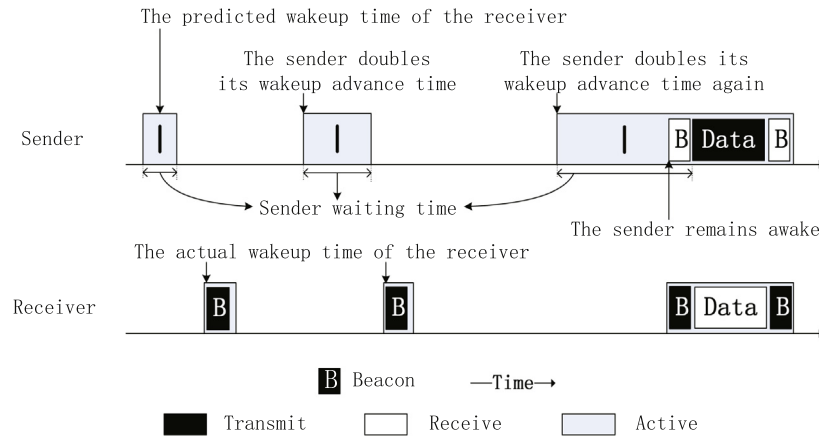


Fig. 6. The operation of exponential advance algorithm.

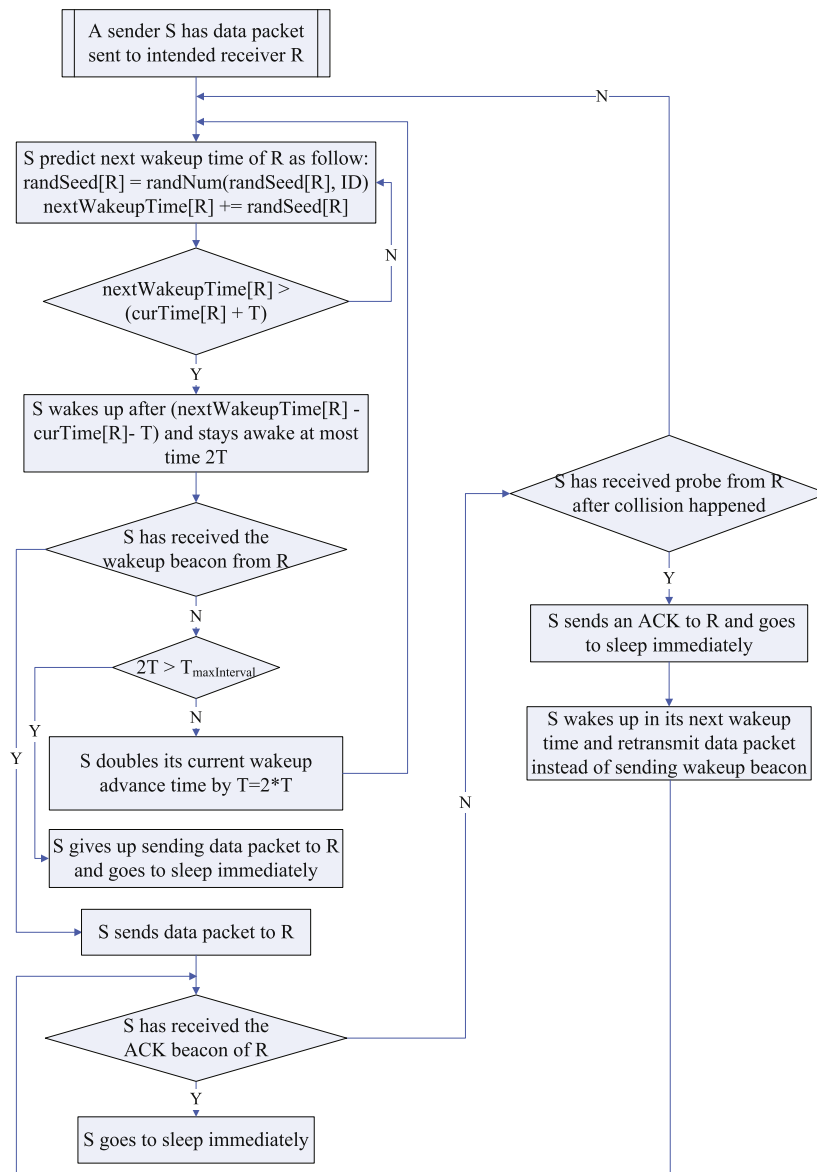


Fig. 7. The prediction-based wakeup algorithm.

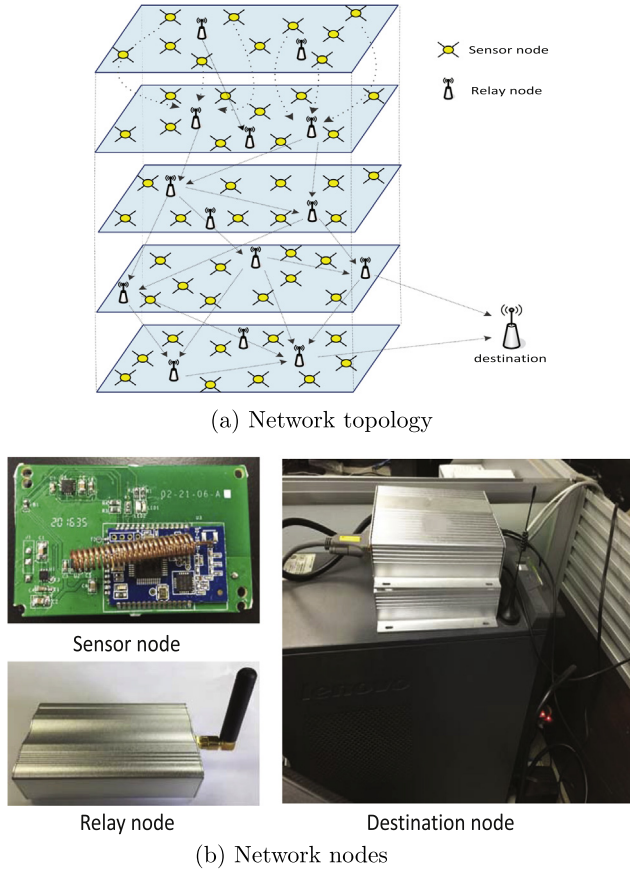


Fig. 8. Sensor nodes are deployed in our experiment.

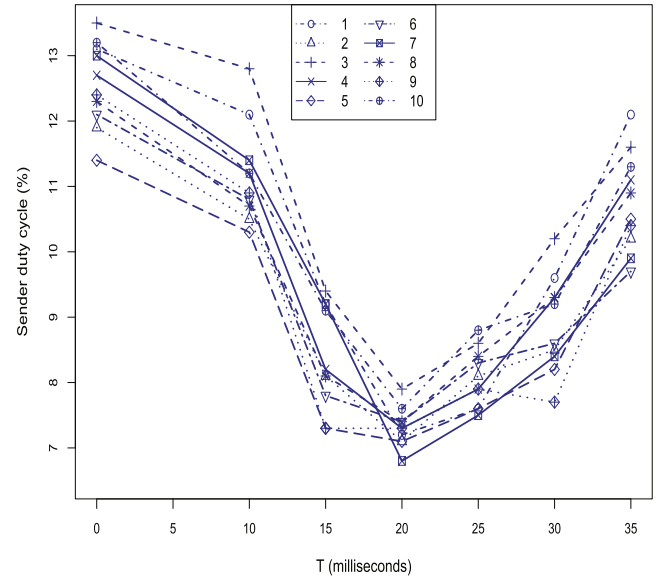
beacon will also increase the sender's duty cycle and prolong packet delivery latency. In other words, the tradeoff between duty cycle and successful reception of wakeup beacon should be considered when choosing the value of  $T$ . According to the experimental result, we set  $T$  to be 20 ms.

Furthermore, we do experiments without wakeup advance mechanism by configuring  $T$  as 0. In Fig. 9a and b, one can see that adding wakeup advance mechanism significantly improves the performance of PBA-MAC. PBA-MAC without this mechanism has the highest duty cycle and packet delay, since the prediction error causes the sender missing some wakeup beacons of the receiver.

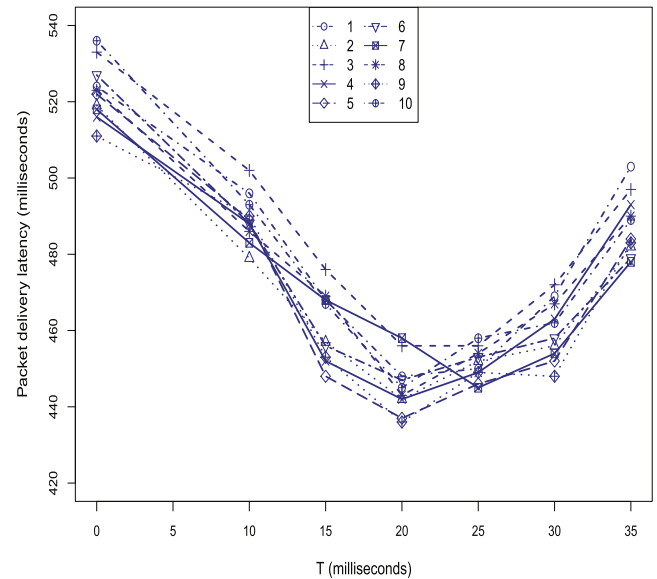
#### 4.2. Experiment results for star networks

We first evaluate PBA-MAC in star topology while multiple sensor nodes send packets to the same receiver simultaneously, which cause wireless collisions. The performance comparison of PBA-MAC is conducted under different traffic loads by changing the number of contending senders. These senders generated one 48 bytes data packet every second on average. The average duty cycle achieved by different protocols are shown in Fig. 10a.

In Fig. 10a, we can see that all of the protocols achieve high performance when there is only one sender. It is worth mentioning that the average duty cycle in PBA-MAC remains under 2% for all traffic loads, while the performance of other protocols degrades quickly as the simultaneously sending nodes increase. When the number of contending senders is set to be 5, the average duty cycle of WiseMAC significantly increases to more than 8%. In Fig. 10b, all protocols can perform a less packet delivery latency when there are only one or two contending senders. However, when there are more than three contending senders, a significant amount of wireless collisions occurred, backoffs can degrade the MAC protocols



(a) The impact of different  $T$  on the sender's duty cycle.

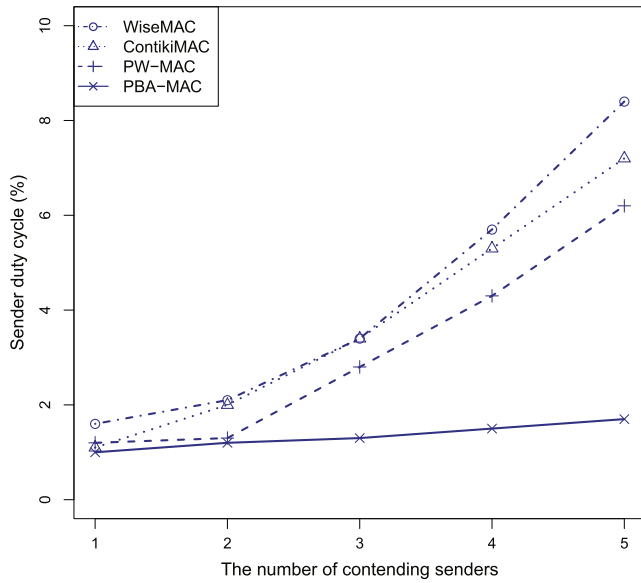


(b) The impact of different  $T$  on the packet delivery latency.

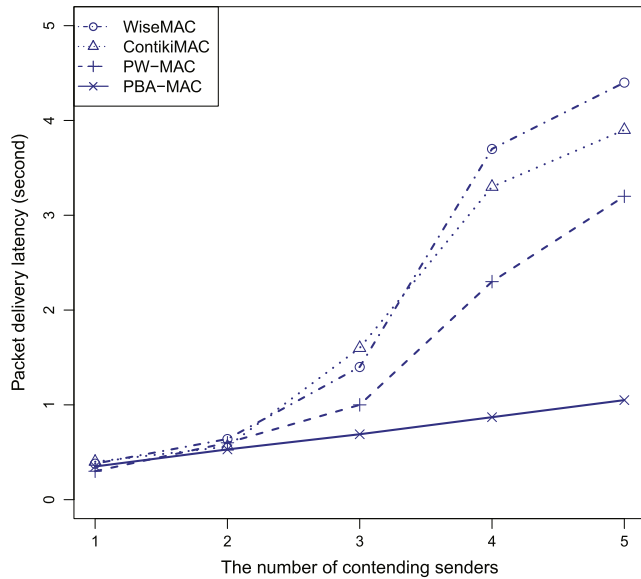
Fig. 9. The average sender duty cycle and packet delivery latency are evaluated under different wakeup advance time  $T$ .

performance. When wireless channel collisions happen, a node must invoke backoff to retransmit packets. Furthermore, persistent collision causes the packet to be retransmitted multiple times, which results in a larger packet delay. In our PBA-MAC, its backcast-based retransmission mechanism enables all contending senders immediately complete packet retransmission in one wakeup cycle under heavy channel contention, which significantly reducing delivery latency.

The reason for PBA-MAC achieving the best performance in Fig. 10 is that PBA-MAC enables nodes to precisely predict wakeup times of neighbors. This prediction method can eliminate unnecessary idle listening for packet transmission. Furthermore, PBA-MAC introduces the exponential expansion algorithm to enable the sender to quickly re-rendezvous with the receiver, greatly reducing



(a) Average sender duty cycle



(b) Average packet delivery latency

**Fig. 10.** The performance of PBA-MAC in star networks.

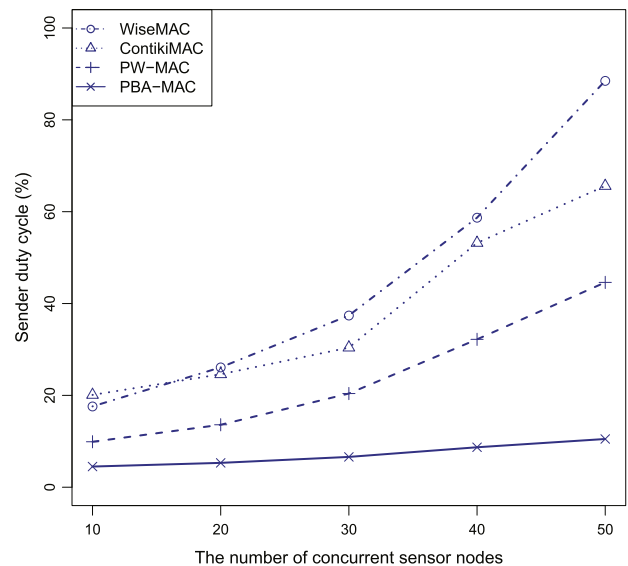
the energy consumption and delivery latency caused by prediction error. PBA-MAC can detect collisions and efficiently retransmit the packets using backcast-based retransmission mechanism even under heavy traffic load. In contrast, the other protocols cannot efficiently resolve collisions other than repeatedly retransmitting the packets, causing senders to stay awake for a long time. These retransmissions may cause more collisions when they interfere with the normal packet transmissions of other senders. However, the backcast-based retransmission mechanism in PBA-MAC enables multiple contending senders to retransmit packets at different time in many-to-one communication manner, which can avoid persistent collisions. Consequently, PBA-MAC can minimize energy consumption and significantly reduce packet delivery latency.

#### 4.3. Experiment results for multihop networks

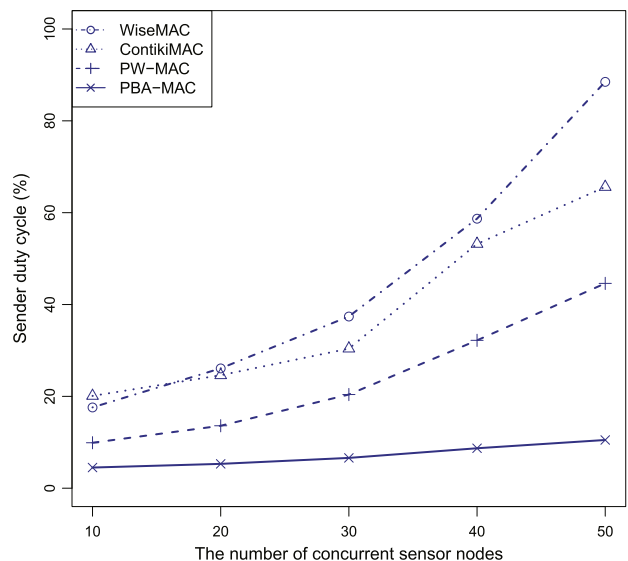
The performance of PBA-MAC for multihop networks is evaluated in this subsection. Fig. 8 illustrates the network topology

and hardware platform. We changed the traffic loads by varying the number of sensor nodes at each floor. The sample data is delivered to the destination by relay nodes, which can conduct concurrent multihop traffic flows within maximum 5 hops. In the experiments, each sensor node periodically sends packet to the destination at different intervals varying from 500 to 1500 ms. The experiment lasted 1000 s.

In the multihop network, data packets are generated by sensor nodes and forwarded to the destination by relay nodes. A multihop path consists of multiple senders and only one destination. Therefore, the duty cycle of senders mainly determines the energy consumption of a multihop path, rather than that of destination. The average duty cycle of senders and packet delivery latency are shown in Fig. 11a and b respectively. We can see that all protocols achieve high performance when there are only ten sensor nodes. The sender duty cycle of WiseMAC and ContikiMAC grows quickly as the number of sensor nodes increases. This is because they are both sender-initiated protocols whose preamble transmission results in large duty cycle. In addition, concurrent preamble trans-



(a) Average sender duty cycle



(b) Average packet delivery latency

**Fig. 11.** The performance of PBA-MAC in multihop networks.



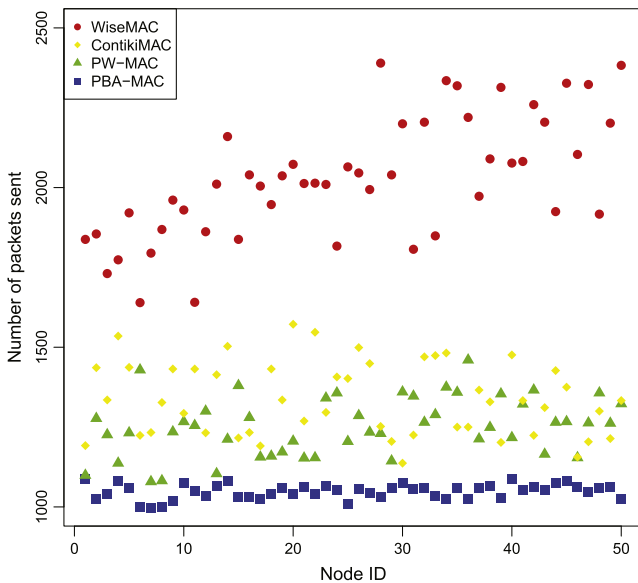


Fig. 12. The number of packets sent by each node for different MAC protocols.

missions of multiple senders within two hops cause severe collisions. Neither WiseMAC nor ContikiMAC proposes an efficient mechanism to resolve wireless collision. PW-MAC performs better since it also enables senders to predict receiver's wakeup times. However, with the number of concurrent traffic flows increases, Binary Exponential Backoff (BEB) strategy in PW-MAC can degrade its performance since all contending senders have to stay awake to conduct retransmissions, resulting in more idle listening and overhearing.

PBA-MAC performs best in terms of duty cycle, and it also achieves the smallest delivery latency. When there are multiple concurrent flows, the packet delivery latency of WiseMAC and ContikiMAC increases rapidly since their preamble transmissions collide seriously under high traffic load. Channel collisions may result in large delay because packets need to be retransmitted hop by hop. Although packets in PW-MAC can be quickly delivered to the destination under light traffic load, the backoff time to retransmit packets significantly increases delivery latency.

We also perform statistics on the number of packets sent by each node, when there are 50 sensor nodes in the experiment, as shown in Fig. 12. We can see that the sensor nodes in PBA-MAC send the minimal number of packets even under heavy traffic load. The significant energy efficiency of PBA-MAC is mainly due to following reasons. First, each node in PBA-MAC uses a sequential pseudo-random numbers as its wakeup intervals. Since different nodes have different wakeup schedules, it is difficult for two nodes consistently waking up at the same time, which can lead to a great reduction in the possibility of collision from hidden terminal. Hence, PBA-MAC greatly reduces the number of packet retransmissions. Second, senders use the exponential advance algorithm to achieve quick and accurate rendezvous with target receiver once missing whose wakeup beacon, thereby minimizing sender idle listening and overhearing. Finally, the backcast-based retransmission mechanism in PBA-MAC can avoid persistent collision and significantly reduces energy consumption and packet delay caused by retransmitting packets.

## 5. Conclusions

In this paper, we have proposed the Prediction-Based Asynchronous MAC protocol (PBA-MAC) for heavy traffic load in WSNs. Senders in PBA-MAC can minimize their energy consumption by

predicting intended receivers' wakeup time. An exponential advance algorithm is introduced to handle the prediction error caused by clock drift and system latency. Moreover, PBA-MAC uses the backcast-based retransmission mechanism to efficiently resolve severe collisions caused by concurrent traffic flows. Finally, the PBA-MAC is implemented and evaluated under different network topologies. The experimental results show that, compared to the latest protocols, PBA-MAC significantly improves network energy efficiency and reduces packet delivery latency even under heavy traffic load.

## Acknowledgements

This work was supported in part by National key R&D plan (Grant Nos. 2016YFC0105002 and 61674162), Shenzhen Key Lab for RF Integrated Circuits, Shenzhen Shared Technology Service Center for Internet of Things, Guangdong government funds (Grant Nos. 2013S046 and 2015B010104005), Shenzhen government funds (Grant Nos. JCYJ20160331192843950 and CXZZ20150601160410510), and Shenzhen Peacock Plan.

## References

- [1] Min X, Wei-Ren S, Chang-Jiang J, Ying Z. Energy efficient clustering algorithm for maximizing lifetime of wireless sensor networks. *AEU-Int J Electron Commun* 2010;64:289–98.
- [2] Singh R, Verma AK. Energy efficient cross layer based adaptive threshold routing protocol for wsn. *AEU-Int J Electron Commun* 2017;72:166–73.
- [3] Chen X, Ma M, Liu A. Dynamic power management and adaptive packet size selection for iot in e-healthcare. *Comput Electric Eng* 2017.
- [4] Hossain A. Equal energy dissipation in wireless sensor network. *AEU-Int J Electron Commun* 2017;71:192–6.
- [5] Alfayez F, Hammoudeh M, Abuarqoub A. A survey on mac protocols for duty-cycled wireless sensor networks. *Procedia Comput Sci* 2015;73:482–9.
- [6] Le Duc T, Le DT, Zalyubovskiy VV, Kim DS, Choo H. Level-based approach for minimum-transmission broadcast in duty-cycled wireless sensor networks. *Pervasive Mob Comput* 2016;27:116–32.
- [7] Ding X, Sun X, Huang C, Wu X. Cluster-level based link redundancy with network coding in duty cycled relay wireless sensor networks. *Comput Netw* 2016;99:15–36.
- [8] Liu X, Zhao S, Liu A, Xiong N, Vasilakos AV. Knowledge-aware proactive nodes selection approach for energy management in internet of things. *Future Gener Comput Syst* 2017.
- [9] Fernandes RF, Brandão D. Proposal of receiver initiated mac protocol for wsn in urban environment using iot. *IFAC-PapersOnLine* 2016;49:102–7.
- [10] Pan G, Feng Q. Performance analysis of a multichannel dynamically aggregative mac protocol for wsn. *J Int Technol* 2014;15:441–50.
- [11] Mickus T, Mitchell P, Clarke T. The emergence mac (e-mac) protocol for wireless sensor networks. *Eng Appl Artif Intell* 2017;62:17–25.
- [12] Duan R, Zhao Q, Zhang H, Zhang Y, Li Z. Modeling and performance analysis of ri-mac under a star topology. *Comput Commun* 2017;104:134–44.
- [13] Asudeh A, Zaruba GV, Das SK. A general model for mac protocol selection in wireless sensor networks. *Ad Hoc Netw* 2016;36:189–202.
- [14] Medani K, Aliouat M, Aliouat Z. Fault tolerant time synchronization using offsets table robust broadcasting protocol for vehicular ad hoc networks. *AEU-Int J Electron Commun* 2017;81:192–204.
- [15] Panigrahi N, Khilar PM. Multi-hop consensus time synchronization algorithm for sparse wireless sensor network: a distributed constraint-based dynamic programming approach. *Ad Hoc Netw* 2017;61:124–38.
- [16] Sahoo PK, Sheu J-P. Design and analysis of collision free mac for wireless sensor networks with or without data retransmission. *J Netw Comput Appl* 2017;80:10–21.
- [17] Le DT, Le Duc T, Zalyubovskiy VV, Kim DS, Choo H. Collision-tolerant broadcast scheduling in duty-cycled wireless sensor networks. *J Parallel Distrib Comput* 2017;100:42–56.
- [18] Ojo M, Adami D, Giordano S. Performance evaluation of energy saving mac protocols in wsn operating systems. In: 2016 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS). p. 1–7.
- [19] Shabaneh AA, Ali AM, Ng CK, Noordin NK, Sali A, Yaacob MH. Review of energy conservation using duty cycling schemes for ieee 802.15.4 wireless sensor network (wsn). *Wireless Pers Commun* 2014;77:589–604.
- [20] Jha MK, Pandey AK, Pal D, Mohan A. An energy-efficient multi-layer mac (ml-mac) protocol for wireless sensor networks. *AEU-Int J Electron Commun* 2011;65:209–16.
- [21] Huang P, Xiao L, Soltani S, Mutka MW, Xi N. The evolution of mac protocols in wireless sensor networks: a survey. *IEEE Commun Surv Tut* 2013;15:101–20.

- [22] Zareei M, Islam AKMM, Zeb A, Baharun S, Komaki S. Mobility-aware timeout medium access control protocol for wireless sensor networks. *AEU-Int J Electron Commun* 2014;68:1000–6.
- [23] Xu L, Zhu X, Dai H, Wu X, Chen G. Towards energy-fairness for broadcast scheduling with minimum delay in low-duty-cycle sensor networks. *Comput Commun* 2016;75:81–96.
- [24] Da Silva RI, MacEdo DF, Nogueira JMS. Duty cycle aware spatial query processing in wireless sensor networks. *Comput Commun* 2013;36:149–61.
- [25] Doudou M, Barcelo-Ordinas JM, Djenouri D, Badache N, Garcia-Vidal J. Performance optimization of duty-cycled mac in delay-energy constrained sensor network under uniform and nonuniform traffic generation. *Int J Commun Syst* 2016;30:3185.
- [26] Du S, Saha AK, Johnson DB. Rmac: A routing-enhanced duty-cycle mac protocol for wireless sensor networks. In: *IEEE INFOCOM*. p. 1478–86.
- [27] Sun Y, Du S, Gurewitz O, Johnson DB. Dw-mac: a low latency, energy efficient demand-wakeup mac protocol for wireless sensor networks. In: *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*. p. 53–62.
- [28] Singh R, Rai BK, Bose SK. A low delay cross-layer contention based synchronous mac protocol for a multi-hop wsn. In: *2016 IEEE Region 10 Conference (TENCON)*. p. 1821–4.
- [29] Khurana M, Thalore R, Raina V, Jha MK. Improved time synchronization in ml-mac for wsn using relay nodes. *AEU-Int J Electron Commun* 2015;69:1622–6.
- [30] Cano C, Bellalta B, Sfairopoulou A, Oliver M. Low energy operation in wsns: a survey of preamble sampling mac protocols. *Comput Netw* 2011;55:3351–63.
- [31] Hill J, Hill J, Culler D. Versatile low power media access for wireless sensor networks. In: *International conference on embedded networked sensor systems*. p. 95–107.
- [32] Buettner M, Yee GV, Anderson E, Han R. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In: *Proceedings of the 4th international conference on Embedded networked sensor systems*. ACM; 2006. p. 307–20.
- [33] Klues K, Hackmann G, Chipara O, Lu C. A component-based architecture for power-efficient media access control in wireless sensor networks. In: *Proceedings of the 5th international conference on Embedded networked sensor systems*. ACM; 2007. p. 59–72.
- [34] Lim S, Kim S, Cho J, An S. Medium access control with an energy-efficient algorithm for wireless sensor networks. In: *IFIP international conference on personal wireless communications*. Springer; 2006. p. 334–43.
- [35] Wong K-J, Arvind D. Speckmac: low-power decentralised mac protocols for low data rate transmissions in specknets. In: *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*. ACM; 2006. p. 71–8.
- [36] Shi X, Stromberg G. Syncwuf: An ultra low-power mac protocol for wireless sensor networks. *IEEE Trans Mob Comput* 2007;6:115–25.
- [37] Lin E-Y, Rabaey JM, Wolisz A. Power-efficient rendez-vous schemes for dense wireless sensor networks. *IEEE international conference on communications*, vol. 7. IEEE; 2004. p. 3769–76.
- [38] Liu S, Fan K-W, Sinha P. Cmac: An energy-efficient mac layer protocol using convergent packet forwarding for wireless sensor networks. *ACM Transa Sensor Netw (TOSN)* 2009;5:29.
- [39] Merlin CJ, Heinzelman WB. Network-aware adaptation of mac scheduling for wireless sensor networks. In: *Proc. of the conference on distributed computing in sensor systems (Poster Session)*, Citeseer; 2007. p. 24–8.
- [40] El-Hoiydi A, Decotignie J-D. Wisemac: An ultra low power mac protocol for multi-hop wireless sensor networks. In: *International symposium on algorithms and experiments for sensor systems, wireless networks and distributed robotics*. Springer; 2004. p. 18–31.
- [41] Dunkels A. The contikimac radio duty cycling protocol. *Swedish Institute of Computer Science* 2012.
- [42] Ko JG, Tsiftes N, Dunkels A, Terzis A. Pragmatic low-power interoperability: Contikimac vs tinyos lpl. In: *2012 9th annual IEEE communications society conference on Sensor, mesh and ad hoc communications and networks (SECON)*. IEEE; 2012. p. 94–6.
- [43] Lin EY. A comprehensive study of power-efficient rendezvous schemes for wireless sensor networks. *IEEE Int Conf Commun* 2005;7:3769–76.
- [44] Liang C-JM, Terzis A, et al. Koala: Ultra-low power data retrieval in wireless sensor networks. In: *Proceedings of the 7th international conference on Information processing in sensor networks*. IEEE Computer Society; 2008. p. 421–32.
- [45] Sun Y, Gurewitz O, Johnson DB. Ri-mac: a receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks. In: *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM; 2008. p. 1–14.
- [46] Jang B, Lim JB, Sichitiu ML. An asynchronous scheduled mac protocol for wireless sensor networks. *Comput Netw* 2013;57:85–98.
- [47] Dutta P, Dawson-Haggerty S, Chen Y, Liang C-JM, Terzis A. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. In: *Proceedings of the 8th ACM conference on embedded networked sensor systems*. ACM; 2010. p. 1–14.
- [48] Dinh T, Kim Y, Gu T, Vasilakos AV. L-mac: A wake-up time self-learning mac protocol for wireless sensor networks. *Comput Netw* 2016;105:33–46.
- [49] Tang L, Sun Y, Gurewitz O, Johnson DB. Pw-mac: An energy-efficient predictive-wakeup mac protocol for wireless sensor networks. In: *IEEE INFOCOM*. p. 1305–13.
- [50] Rhee I, Warrier A, Aia M, Min J, Sichitiu ML. Z-mac: a hybrid mac for wireless sensor networks. *IEEE/ACM Trans Netw (TON)* 2008;16:511–24.
- [51] Song W-Z, Huang R, Shirazi B, LaHusen R. Treemac: Localized tdma mac protocol for real-time high-data-rate sensor networks. *Pervasive Mob Comput* 2009;5:750–65.
- [52] Cao H, Parker KW, Arora A. O-mac: a receiver centric power management protocol. In: *Proceedings of the 2006 IEEE international conference on network protocols*. IEEE; 2006. p. 311–20.
- [53] Aoudia FA, Gautier M, Berder O. Opwum: Opportunistic mac protocol leveraging wake-up receivers in wsns. *J Sensors* 2015;2016.
- [54] Omar HA, Zhuang W, Li L. Vemac: A tdma-based mac protocol for reliable broadcast in vanets. *IEEE Trans Mob Comput* 2013;12:1724–36.
- [55] Chan YL, Hong IC, Gang UH, Doh Y, Park N. Performance modeling and analysis of ieee 802.15.4 slotted csma/ca protocol with ack mode. *AEU – Int J Electron Commun* 2011;65:123–31.
- [56] Dutta P, Musaloiu-e R, Stoica I, Terzis A. Wireless ack collisions not considered harmful. In: *Proceedings of the 7th ACM workshop on hot topics in networks (HotNets-VII)*. p. 1–6.
- [57] Fontaine C. *Linear congruential generator*. Springer US; 2005.
- [58] Knuth DE. *The linear congruential method*. *Art Comput Programm* 1997;2:10–26.