Bo C, Li XY, Wang Y, Xiao B. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY : 1–
Mon. Year

SA-MAC: Self-stabilizing Adaptive MAC Protocol for Wireless Sensor Networks

Cheng Bo[1] (波澄), Xiang-Yang Li[2] (李向阳), Yu Wang[1] (王昱), Bo Xiao[3] (肖波)

[1] *University of North Carolina at Charlotte, Computer Science, Charlotte, North Carolina, United States*
[2] *Illinois Institute of Technology, Computer Science, Chicago, Illinois, United States*
[3] *The Aeronautical University of PLA, Jinzhou, China*

E-mail: frank_bocheng@hotmail.com, xiangyang.li@gmail.com; wangyu@gmail.com; xxbb@163.com

Received month day, year

Abstract A common method of prolonging the lifetime of wireless sensor networks is to use low power duty cycling protocol. Existing protocols consist of two categories: *sender-initiated*, and *receiver-initiated*. In this paper, we present **SA-MAC**, a Self-stabilizing Adaptive MAC protocol for wireless sensor networks. SA-MAC dynamically adjusts the transmission time-slot, waking up time-slot, and packet detection patten according to current network working condition, such as packet length and wake-up patterns of neighboring nodes. In the long run, every sensor node will find its own transmission phase so that the network will enter a stable stage when the network load and qualities are static. We conduct extensive experiments to evaluate the energy consumption, packet reception rate of SA-MAC in real sensor networking systems. Our results indicate that SA-MAC outperforms other existing protocols.

Keywords Duty Cycling Protocol, sender-initiated, receiver-initiated, SA-MAC

1 Introduction

Reducing the energy consumption in wireless sensor networks attracts most of attention in recent years. In traditional WSNs, the most energy-consuming component is demonstrated

The research of Li is partially supported by NSF CNS-0832120, NSF CNS-1035894, NSF ECCS-1247944, NSF ECCS-1343306, National Natural Science Foundation of China under Grant No. 61170216, No. 61228202.

to be the wireless transceiver. In order to save as much energy as possible, the transceiver should be turned off completely when the sensor node does not transmit packet. One of the most popular mechanisms for achieving low energy consumption in WSNs is duty-cycle. In this mechanism, each sensor nodes will wake up periodically to transmit or receive the data and go back to sleep when the time expires. And the cycle time of one duty cycle period is its sleep duration plus active duration. For example, a WSN operating in a 5% duty cycle, each sensor node will turn on its transceiver for only 5% of the cycle time. According to this scheme, the total energy consumption of a sensor node depends on the exact time phase of both sleep and wake up, and the corresponding duration. Although many protocols have been proposed to reduce the energy consumption, it is still a challenge to optimize the protocols to provide higher throughput and lower delay for energy-constrained low duty-cycle wireless sensor networks.

Currently, researchers have proposed contention-based duty cycle MAC protocols, and the idea of periodic wake-up scheme has been introduced into existing protocols, such as B-MAC [1], S-MAC [2], T-MAC [3], and X-MAC [4]. The main challenge in such duty-cycle MAC protocols is to reduce the network energy consumption while maintaining the high throughput and packet delivery ratio. Contiki-MAC [5], which has been proposed recently, is a sender-initiated MAC protocol. In such network, the sensor nodes could satisfy the demands of network communication under the circumstance that the radio is turned off for about 99% of one period.

Each sensor node works individually according to its own duty cycle. Currently sensor nodes play both sender and receiver roles in the network. However, from the perspective of sender, it does not know exactly when the receiver will wake up to receive the packet. A feasible solution is to keep on listening the channels until it knows that the intended receiver wakes up, which may lead to packet loss and huge energy consumption. Another approach is receiver initiated, in which the receiver stays awake to listen to any income packets.

In this work, we study the impact of various parameters in the MAC protocols that are often neglected. We show that by carefully setting the values of these parameters in accordance to the network environment, the network performances (e.g., energy consumption, and packet delivery ratio) are improved. Based on this investigation, we propose a new MAC protocol called SA-MAC, which is a sender-initiated Self-stabilizing Adaptive MAC protocol. SA-MAC takes advantage of ContikiMAC, which is the latest low-power duty cycling MAC protocol, and sensor nodes will adjust their transmission and packet detection patten dynamically according to current network condition, such as packet length and wake-up mechanism. As nodes dynamically adjust their wake-up phases based on the working patterns of its neighbors, we show that, under our protocol, the network will evolve to a TDMA-like protocol when the environment remains stable.

To evaluate the performances of our protocol, we first conduct extensive simulations to find guidance on optimal parameter settings. Based on this, we then implement the SA-MAC and conduct extensive testbed evaluations on the performances. Our results show that SA-MAC indeed outperforms the existing protocols, e.g., the average energy consumption is reduced by from 5% to about 50%.

The rest of the paper is organized as follows. Section 2 will review the related work. Section 3 will describe the design of SA-MAC protocol, including the theoretical analysis and the system model. Section 4 presents the implementation of SA-MAC protocol in Contiki 2.5, and the evaluation in real system. We conclude the paper in Section 5.

2 Related Works

The MAC protocols in sensor networks have been well-studied recently, and many duty-cycle MAC protocols have been presented. Generally, existing radio duty cycling mechanisms could be divided into two main categories: *synchronous* and *asynchronous*. The former mechanism is based on the condition that the nodes are initially synchronized with each other, the waking up period and sleep period are concurrent, while the latter does not.

Examples of synchronous protocols include S-MAC [2], T-MAC [3], TSMP [6], R-MAC [7], and DW-MAC [8]. Nodes in this kind of networks are synchronized with their neighbors in order to coordinate their active and sleep periods. This mechanism could reduce the time for idle listening, because sensor nodes only exchange packets within their common active timeslots, and sleep together most of the time. However, such mechanisms could introduce extra overhead and complexity. In both S-MAC [2] and T-MAC [3], sensor nodes wake up in scheduled manner, exchanging synchronization and schedule information with neighbors to ensure they wake up simultaneously. The S-MAC [2] is based on RTS-CTS mechanism, which will increase extra communication cost to some extent. While T-MAC [3] will reduce the period of active as long as the channel is sensed to be idle. If no packet to be transmitted the node will turn off the radio and go to sleep period. If a data packet is received, the node will stay awake to make sure that no further data is received or the active periods finish. Although T-MAC costs much less energy than S-MAC, the throughput is proved to be reduced and the latency is increased. TSMP [6] divides the time into 10 slots, and the transceiver will be turned on at the beginning of each time slot to check the activity in the channel. The radio will be kept on if incoming packet is detected, while turn off when the time is expired or no activity is sensed.

Different from synchronous protocols, the asynchronous mechanisms reduce the overhead brought about by the required time synchronization, and researchers have explored many asynchronous MAC protocols, such as B-MAC [1], X-MAC [4], WiseMAC [9], RI-MAC [10], LPP [11], and ContikiMAC [5]. Some of these protocols employ low power listening (LPL), in which each sender will transmit a preamble to set the communication with receivers initially (sender-initiated mechanism), and such preamble will usually last at least the sleeping period of receivers. If the a receiver wakes up and detects the preamble successfully, it will keep active to receive the packets. Although the LPL is energy-efficient and simple, the long period of preamble in LPL still contains several drawbacks. The receiver have to stay active for the period of preamble before starting receiving data packet and returning ACK, which will lead to a tremendous packet delivery latency. In a multi-hop network, the latency could be accumulated to become huge. In addition, if some none-intended receiver finds out that it is not the target receiver after having detected the whole period of preamble, there will be a huge energy waste.

B-MAC [1] has high performance in light traffic because of its very short period of time in detecting the channel activity at each scheduled wake-up time, but will cause high overhearing when a node is in its wake-up period. X-MAC [4] solves this kind of drawback by providing a strobed preamble, which consists of sequence of short preambles prior to data transmission. But this protocol still could not handle a wide range of traffic loads more efficiently. The WiseMAC [9] is similar to the previous two, but the difference exists in the length of wake-up preamble. The wake-up preamble is shortened by learning the sampling of the schedules of its neighbors, and scheduling its own transmission. However, the preamble sampling techniques might lead to the possibility of simultaneous transmission from hidden sensor nodes. The latest sender-initiated MAC protocol is called ContikiMAC [5], it provides a more significant energy-efficient wake-up mechanism by introducing the fast sleep and phase-optimization mechanism. It reduce the wake-up interval as low as 125ms.

As to the receiver-initiated mechanism, the receiver will start up the communication. One of the receiver-initiated mechanisms is LP-P [11], which is designed for establishing reliable download bulk data connection from all sensor nodes. The node turns on its transceiver when it intends to transmit a packet. As long as it detects a probe from the receiver, it will start sending its data packet. In RI-MAC [10], sender will wait silently until receiver transmits a short beacon. The occupancy of medium will decrease because of no preamble message, which will allocate more time for data transmission. The neighbor nodes coordinate with each other through these short beacon, and receiver will adjust the channel utilization according to current traffic load, which result in achieving the high throughput, packet delivery ratio and low energy cost in a large scale sensor network.

3 SA-MAC Design

In this section, we will first describe the basic idea of the SA-MAC protocol briefly, and establish a model of the energy consumption of this protocol. We will then analyze the performance according to different parameters.

3.1 Protocol Overview

The SA-MAC is a low duty cycle MAC protocol, and it is designed based on ContikiMAC, but with more efficiency and less energy consumption. Compared with ContikiMAC, whose wireless transmission pattern is fixed, our SA-MAC protocol adjusts the packet transmission according to current network condition, such as packet length, network collision, and delay.

ContikiMAC is an energy-efficient sender-initiated protocol, the senders are responsible for building connections with receivers. Every node consists of two independent *protothreads* which are responsible for transmitting and listening respectively. Once the time of transmission expires, and the sender will wake up to initialize to active transmission. During this period, the sender will broadcast the *same* packet repeatedly until it receives a link layer acknowledge from the intended receiver. In this case, we say that *the sender is hit by the receiver*. After the sender is hit, it will record the reference time when its packet being received, the receiver's ID, and calculate the *phase time* in its duty-cycle. In the following cycle time, the sender will wake up at the modified phase time. For example, if the sender receives the ACK from the receiver after its i-th packet at its initial cycle, and the sender will wake up at the time when $(i-1)$th packer was sent in the following cycles, which will save more energy by not sending useless packets.

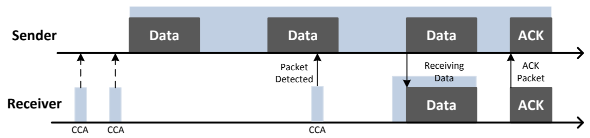


Fig. 1. ContikiMAC Mechanism

The *thread of listening* is also triggered periodically. The most efficient and energy saving scheme is to employ Clear Channel Assessment (CCA) to check the activity in the channel, as shown in Figure 1. The adoption of CCA is based the assumption that a packet will carry a signal intensity, which is Received Signal Strength Indicator (RSSI). The transmission energy is assumed to be high enough to exceed a specific threshold for easy detection and ignore the extraneous noise simultaneously [12]. Before the receiver starts receiving, it will turn on the radio to detect the RSSI value in the channel. If the RSSI value is above the clear channel assessment threshold (CCAT), the indicator returns *negative*, which demonstrates the channel is occupied. By contrast, if the RSSI value is below the threshold, the channel is clear to use. ContikiMAC employs two successive CCAs to check the existence of a packet transmission in the channel, and decide the choice of other parameters in the protocol, such as the interval between each packet transmission. When the receiver detects a packet in the channel, it is supposed to receive the following packet. At this time, the transceiver could be turned off at least d milliseconds before being turned on again for receiving, where d is the time interval of two successive packets that sender transmits in its active period. A link layer acknowledge will also be sent out to the sender after successfully receiving the packet. In addition, CCA detection is also operated before the sender sends packet to prevent collision.

3.2 MAC Protocol Model

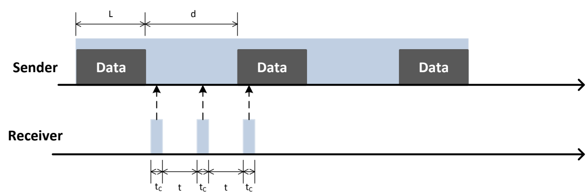


Fig. 2. The model and parameter of a MAC Protocol.

The main purpose of low duty cycle MAC protocol in WSNs is to reduce the energy consumption while assuring the wireless transmission quality. According to the past experience, as to one sensor node, most energy are consumed in the radio module, with $17.4mA$ in transmission and $19.7mA$ in receiving [13]. Take TelosB sensor nodes as an example, the

d	the time interval between two successive packets.
L	the time for transmitting a packet.
t	the time interval between two successive CCA.
t_c	the time consumed by each CCA detection.
K_s	number of times the same packet is repeatedly transmitted
K_r	the number of consecutive CCA detections by receiver

Table 1. Parameters used in MAC protocol analysis.

energy consumption of CC2420 in listening is $63mW$ and $60mW$ for transmitting.

First, we establish a model to analyze the power consumption in our MAC protocol, shown in Figure 2. Table 1 summarizes the parameters that we will optimize to reduce the energy consumption. The goal of establishing the model is to find the suitable parameter in the protocol to minimize the energy consumption. The model could be discussed in two cases, according to the length of the time interval between two packets and the time of CCA to checking the activity of the channel (denoted as t_r, where $t_r = K_R(t_c+t)-t$). For convenience, let $t_s = K_S(L+d)-d$, which presents the active time or waking up time of the sender.

3.2.1 Case 1: $d < t_r$

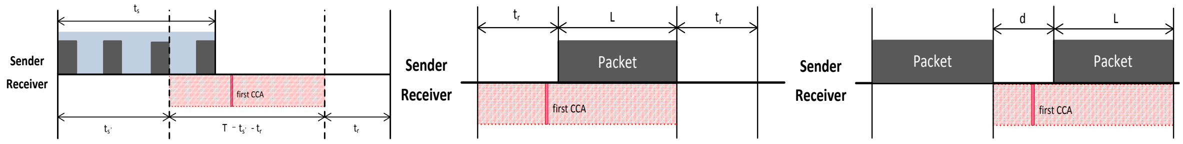
In this case, we have to consider all possible cases of detecting the packet, and calculate the expected energy consumption.

First of all, we should take the possibility of all packets not being detected by all CCAs and no connection is established in the cycle time. In this condition, all CCAs are located after the (K_S-1)th packet transmission and outside the active period of sender, as shown in Figure 3(a). So the possibility of not detecting any packet could be calculated as:

$$P_0 = \frac{T - t'_s - t_r}{T} \tag{1}$$

where $t'_s = (K_S-1)(L+d)-d)$. The reason for the equation here using t'_s rather than t_s is because according to the MAC protocol, the sender will start transmitting packet as soon as waking up, however even if the last packet is detected, the connection still cannot be established.

If the first packet is detected, which means the packet is sensed by the intended receiver as soon as the sender wakes up. Under this circumstance, the boundary condition would be the last CCA hits the beginning of the packet



(a) No packet being detected (b) The first packet being detected (c) The following packet being detected

Fig. 3. Packet Detection Condition in Sender

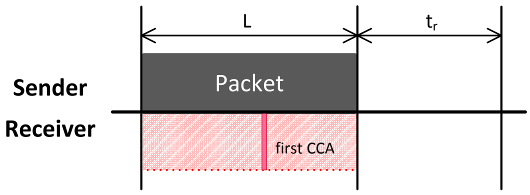
and the first CCA hit the ending of the packet, as shown in Figure 3(b). So the possibility could be easily presented as:

$$P_1 = \frac{t_r + L}{T} \tag{2}$$

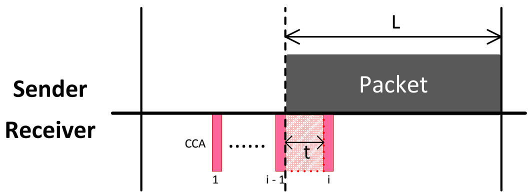
As to the following $K_S - 2$ packets, the beginning of CCA falls in the period of $d + L$ will lead to being detected, as shown in Figure 3(c). Thus the probability in this condition is

$$P_i = \frac{d + L}{T} \tag{3}$$

where i is from 2 to K_S.



(a) First CCA detects the packet



(b) Following CCA detects the packet

Fig. 4. Packet detection cases at receiver

We now analyze the cases at the receiver side. CCA detection will be activated when the receiver wakes up. And there are also three conditions for CCA detecting packets. First, if the receiver does not sense any available packet in the channel, the condition is absolutely the same as the sender (see Figure 3(a)), and the probability Q_0 is also same to the sender P_0.

$$Q_0 = \frac{T - t_s' - t_r}{T} \tag{4}$$

If the first CCA successfully detects the packet as soon as waking up, such CCA must fall into the period of one packet transmission, as shown in Figure 4(a). The possibility in this case will be:

$$Q_1 = \frac{L}{T} \tag{5}$$

For the condition that if the i-th CCA could detect the packet, such CCA only could locate at the range of every beginning of the packet, and the duration will be t. We could check this condition in Figure 4(b). In addition, as the range of d is smaller than t_r, at most $2 + \lfloor \frac{d-t_c}{t_c+t} \rfloor$ CCAs could be used to detect the packets. Therefore, the possibility of the following CCAs detect the packet is

$$Q_i = \frac{t}{T} \quad i = 2, 3, \cdots, 2 + K_C \tag{6}$$

where $K_C = \lfloor \frac{d-t_c}{t_c+t} \rfloor$.

3.2.2 Case 2: $d \geq t_r$

When the time interval of two successive packets is larger than the active period of receiver, the results will be different. For the sender, there are only two conditions needed to be taken into account. One is no packet being detected, and the other is successfully a packet is successfully detected.

For the former condition, if the CCAs from the receiver is activated during the sleeping period and the time interval of two continuous packets, no packets will be detected. The possibility for this condition is

$$P_0 = \frac{T - K_S(L + t_r)}{T} \tag{7}$$

The possibility of the i-th packet being detected is

$$P_i = \frac{t_r + L}{T} \quad i = 1, 2, \cdots, K_S \tag{8}$$

At this time, receiver also has to be considered for three conditions: not detecting, first CCA detects, and the following CCA. Then we have

$$\begin{cases} Q_0 = P_0 = \frac{T-K_S(L+t_r)}{T} \\ Q_1 = \frac{L}{T} \\ Q_i = \frac{t}{T}, (i = 2 \ \ to \ \ K_S) \end{cases} \tag{9}$$

3.2.3 Overall energy consumption

For both of these two cases, we will calculate the expected energy consumption for sender and receiver respectively, and in addition we could get the expected total energy consumption in this MAC protocol. Suppose the power consumptions for transmitting, receiving, and CCA detecting packet are denoted as E_s, E_r, and E_c respectively.

The expected energy consumption of sender could be calculated as:

$$\bar{E}_S = \sum_{i=0}^{K_S} P_i \cdot E_{S_i} \tag{10}$$

For both $d < t_r$ and $d > t_r$, if no packet is detected, the packet will be transmitted K_S times in vain. For every other packet except the last one, when the i-th packet is detected, the packet will be transmitted $(i + 1)$ times. As to the last packet, even if it is detected by the CCA from the receiver, the connection still cannot be established. Therefore, the expected value is $\bar{E}_S = P_0 \cdot K_S \cdot L \cdot E_s + L \cdot E_s \cdot \sum_{i=2}^{K_S} i \cdot P_i + P_i \cdot K_S \cdot L \cdot E_s$.

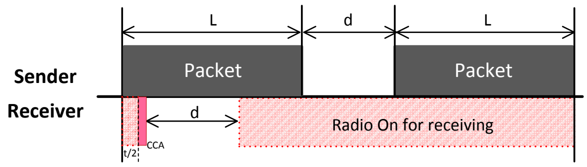


Fig. 5. Average duration for receiving

On the other hand, receiver will start receiving packet when detecting a packet in the channel. Since the time interval of two continuous packet is presented as d, in our protocol, once the receiver detects the packet through C-CA, the radio will be turned off at least for "d" milliseconds before it starts receiving again. For every receiver, if the i-th CCA detected the packet, the total average duration for the receiver turning on the radio to receive packet is (illustrated in Figure 5) $t_l = 2L - \frac{t}{2} - t_c$. When $d < t_r$, the expected energy consumption of receiver is $\bar{E}_R = \sum_{i=0}^{2+K_C} Q_i \cdot E_{R_i} = Q_0 \cdot K_R \cdot t_c \cdot E_c + \sum_{i=1}^{2+K_C} Q_i \cdot (i \cdot t_c \cdot E_c + t_l \cdot E_r)$. On the contrary, when $d > t_r$, the expected energy consumption of receiver is $\bar{E}_R = \sum_{i=0}^{K_R} Q_i \cdot E_{R_i} = Q_0 \cdot K_R \cdot t_c \cdot E_c + \sum_{i=1}^{K_R} Q_i \cdot (i \cdot t_c \cdot E_c + t_l \cdot E_r)$.

Since CCA could only detect the activity of the channel rather than parsing the packet, once receiver senses the packet, it will turn on the radio for next incoming packet even if it is not supposed to be the destination of the packet. Therefore, the expected energy consumption of one cycle time could be: $\bar{E}_1 = \bar{E}_S + N\bar{E}_R$, where N indicates the number of neighbor nodes of the sender.

3.3 Gross Energy Consumption

In ContikiMAC, a phase-lock mechanism is adopted to optimize the energy consumption. Suppose both sender and receiver will wake up periodically and stably, and the sender could learn the intended receiver's phase through the time relationship between its packet transmission and the link layer acknowledge reception. Once sender receives the link layer acknowledge, it will get the information of which packet was received successfully by which receiver, as well as the waking up phase of the receiver. In the following cycle times, the sender could postpone its waking up time to the very moment before the receiver wakes up, as shown in Figure 6. Therefore, energy will be saved.

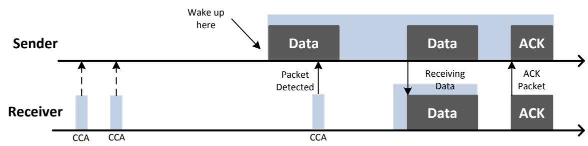


Fig. 6. Postpone waking up time: phase-lock.

According to our previous experience in designing practical WSNs, the randomly deployment and dynamic routing will lead to the possibility that the connection between sender and receiver could not be established. One receiver could have multiple senders as neighbor, some may not transmit their packets successfully in the initial cycle, or the collision prevents the normal communication. Under this circumstance, the sender will re-schedule its waking up time randomly in the next cycle until it finds the phase of the receiver.

Suppose the network will run M cycle times, and the sender finds the receiver's phase at its first cycle, then the total expected energy consumption is

$$E_{(1)} = E_1 + (M-1)E_t \quad (11)$$

where E_t indicates the energy consumption of every following cycle. In these cycles, sender will have to transmit only two packets as long as the phase is still fixed.

$$E_t = 2 \cdot E_S \cdot L + \bar{E}_R \quad (12)$$

However, the total expected energy consumption for the M cycles is difficult to achieve. If the sender missed the receiver's phase in its first cycle but the second, the expected total energy consumption is $E_{(2)} = P_0 \cdot E_1 + (M-2)E_t$. Thus , if the packet from the sender is received in the i-th cycle, the total expected energy consumption could be expressed as $E_{(i)} = P_0^{i-1} \cdot E_1 + (M-i)E_t$, for $i \in [1, M]$. Therefore, the gross expected energy consumption is

$$E = \sum_{i=1}^{M} E_{(i)} = \frac{E_1}{1-P_0} + \frac{M \cdot (M-1)}{2} \cdot E_t \quad (13)$$

3.4 Energy Consumption Minimization

According to the model, the main factors influence the energy consumption are the number of transmissions, duration of each packet transmission, number of CCA detections. An naive strategy in this protocol is to reduce these three parameters. However this method could increase the chance of packet lost and delay. More constraints have to be added into the model, such as bounding both the time interval of two successive CCA detections, and two successive packet transmissions.

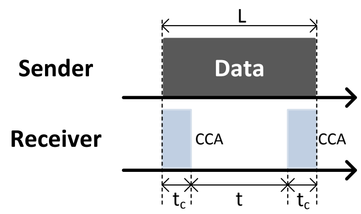


Fig. 7. The minimum length of packet in Contiki.

First of all, the time interval of CCA should be smaller than the packet length. As shown in Figure 7, $2t_c + t < L$ to prevent the packet from falling between two CCAs. In addition, according to the datasheet of CC2420, the maximum packet buffer is 128 bytes [13], with an effective data rate 256 kbps, thus the maximum packet length (denoted as transmission time) will be $L_{max} = 128 * 8/256 = 4$ milliseconds. On the other hand, the duration of CCA detection is 0.192 milliseconds according to the datasheet of CC2420, then the time t is no larger than $L_{max} - 2t_c = 4 - 2*0.192 = 3.616$ milliseconds.

Another constrain we have to take into account is the packet reception rate. According to the protocol above, the sender will repeatedly transmit its packet during its active time, and will wake up again in the next cycle time if the phase is not hit. One reason for packet loss is that almost every sensor node will act as both source node to generate data, and relay node to forward data for others. The traffic load of some node will be large enough, which may result that packets could not be delivered to the intended receiver in time. The upper bound of the constrain is that one packet should be delivered within one sampling cycle. In Figure 8, assume that sensor R will transmit its sampled data to the next hop, which is sensor T. Meanwhile, sensor R has to forward the packet from sensors A, B and C to T as well. However, sensors A, B and C all may have their own subtrees, the number of packets that sensor R has to send could be much larger than its neighbor count.

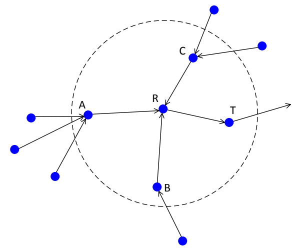


Fig. 8. Packet transmission for one sensor node.

Assume that each sensor node will produce one data packet in one sampling period. Suppose one sensor node has N one-hop neighbors, the number of data packets in one sampling period it has to transmit is at least $2N$. As mentioned in the previous section, sender will wake up periodically to try to establish connection with receiver until one successful transmission. Suppose S be the sampling cycle, T is one cycle time of transmission, K is the intended transmission times within one sample cycle, e is the efficiency of collision avoidance in CSMA/CA. The expected time for successful transmission should be smaller than timeslot allocated to one packet, as shown in Equation 14. Consequently,

$$\sum K \cdot P_K < \frac{S}{e \cdot 2N \cdot T} \qquad (14)$$

where P_K indicates the possibility of successful transmission in the K-th cycle. In this case, we have $P_K = P_0^{K-1}(1 - P_0)$, and

$$F(K_R, t, K_S, L, d) = \frac{E_1}{1 - P_0} \qquad (15)$$

According to Equation 13, given a WSN, the second part of the above equation will be constant. Then we need minimize the first part, as shown in Equation (15). Therefore, the final expected energy consumption is a function based on K_R. t, K_S, L, and d. In order to get the theoretical minimum energy consumption in the MAC protocol, we will adjust the parameters based on current network condition. In different scenarios, different work loads will lead to distinct packet lengths, and adjust the number of packet transmission as well as the interval will reduce unnecessary energy consumption. What's more, the duration of transmitting one packet is much longer than CCA detection. Therefore, it is feasible to prolong the time interval between two successive packet transmissions according to the actual packet length and increasing the number of CCA checking. Based on the energy consumption model established in the previous section, increasing times of CCA checking will simultaneously reduce the possibility of no packet being detected. The new challenge is to find the suitable parameter while satisfying the energy consumption model.

4 Implementation and Evaluation

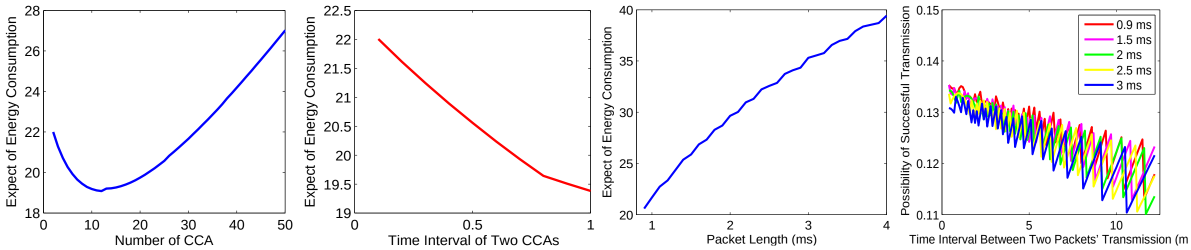
We implement the SA-MAC protocol in Contiki 2.5 to evaluate the performance in real sensor networking systems. Before that, we first calculate the theoretical expected energy consumption based on the model discussed in previous section.

4.1 Simulation Results

In order to compute the expected energy consumption, the wake up frequency of the model is set to be $8Hz$, and the waking up time interval is set to be $125ms$. In addition, we put a delay constrain to compute the feasible parameter. We assume the packet has to be received within five duty cycles.

4.1.1 Case 1

In the first case, when $d < t_r$, the time interval between two CCAs should be no longer than the maximum packet length. According to the data transmission rate, the maximum buffer for packet is 4 seconds, then the range of two successive CCAs should be fall into 0.1 and $4-2*0.192 = 3.626$, and meanwhile the number of CCAs is set to be no larger than 50. The other condition is $t < L$. Figure 9(a) presents the result of impact of increasing number of CCAs on the expected energy consumption. From the curve in this figure, the minimum expected energy consumption emerges under the condition that the number of CCAs is 12. Increasing the number of CCA in detecting data in the chan-



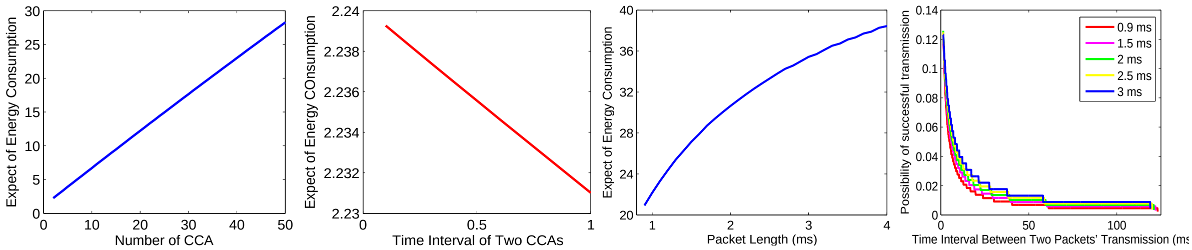
(a) The expect of energy consumption vs. number of CCA

(b) The expect of energy consumption vs. time interval between two successive CCAs

(c) The expect of energy consumption vs. packet length

(d) The expect of energy consumption vs. time interval between two successive packets

Fig. 9. Case 1



(a) The expect of energy consumption vs. number of CCA

(b) The expect of energy consumption vs. time interval between two successive CCAs

(c) The expect of energy consumption vs. packet length

(d) The expect of energy consumption vs. time interval between two successive packets

Fig. 10. Case 2

nel will increase unnecessary energy consumption.

To calculate the best choice of the number of CCAs, we put the parameter into the model again to study the influence of the time interval between two CCAs. The result is shown in Figure 9(b). Although the expected energy consumption increases with the increase of the time interval at the very beginning, the simulation shows that the expected energy consumption will be much lower when the length of time interval of two CCAs decreases. Because of the constraint of t we mentioned before, the preferred time interval between two successive CCAs is $L - 2t$. We also calculate the relationship between the packet length and the expected the energy consumption, it is obvious that with the packet length increasing, the energy consumption grows simultaneously, the result is presented in Figure 9(c).

To get the theoretical feasible parameter of K_R, t, and the effort of the packet length, we should calculate the number of packet transmission (K_S), and the time interval of two successive packets (d). In this step, we calculate the possibility of packet being successfully detected under different time intervals of two packets' transmission, as shown in Figure 9(d). We simulate 5 different lengths of packet, and discover that there is a tradeoff between the d and expected energy consumption: although prolonging the time interval of two successive packets' transmission will increase the possibility of packet detection, the energy consumption will go up at the same time.

4.1.2 Case 2

The second case is under the condition that $d > t_r$, and we conduct the similar simulation. We funf that the results are different. The range of time interval between two CCAs is from $0.1ms$ to $1ms$, and meanwhile the number of CCAs is also no larger than 50. The Figure 10(a) presents the result of impact of increasing the number of CCA to the expected energy consumption. Different from the previous case, the expected energy consumption will rise with the number of CCAs increases: the less times of CCA detection, the less energy will be consumed. The second result in Figure 10(b) also indicates that the larger the time interval between two CCAs, the less energy will cost. According to the initial simulation, an obvious result is that the time interval of two CCAs is related to the total energy consumption. The relationships between expected energy consumption and packet length, and possibility of packets being detected and the time interval of two packets' transmission are conducted. The results, see Figure 10(c) and Figure 10(d), are similar to the previous case.

4.2 Feasible Parameter Selection

The simulation results for both two cases mentioned above indicate four basic phenomena:

1. The expected energy consumption of case 1 is larger than that of case 2.
2. The more times of CCA detection, the more power the network has to consume.
3. The energy consumption is obviously dependent on the packet length and number of packet transmission.
4. With the time interval of two successive packets' transmission increasing, the energy consumption reduces, but the probability of packet not being successfully sensed increases simultaneously.

Therefore, the total energy consumption and network performance depend K_R, t, L, and d. In addition, we also have to consider the tradeoff between energy consumption and possibility of packet detection. In order to get more optimal results, the parameters should be adjusted dynamically according to current network condition, rather than fixing the working pattern.

In this case, we set the K_R to be 2 to reduce the unnecessary energy consumption in CCA detection. As far as network starts working, the MAC layer will get the information of the packet length from the upper layer. The time between the two CCAs $t = L - 2t_r$, which will guarantee that packet could be sensed within two CCAs. In addition, in order to reduce the power consumption while maintaining acceptable possibility of packet detection, in our protocol, d is set to be equal to t. However, senders will have to receive possible Link Layer Acknowledge from receiver, then d should be larger than the time that receiver prepares to send an ACK plus the time for successfully detecting an ACK. According to the specification of IEEE 802.15.4, the time of receiver preparing ACK will cost 12 symbols, which is $12*4/256 = 0.192ms$. The time for successfully detecting an ACK in IEEE 802.15.4 is 10 symbols, which takes $40/250 = 0.16ms$. Then the lower bound of d and t is $0.192 + 0.16 = 0.352ms$.

4.3 Performance Evaluation in Testbed

The simulation results offer us a feasible parameter for the protocol. In this section, we will evaluate the performance of the SA-MAC protocol through a long-term experiment. In this experiment, a testbed of over 45 telosb [14] sensor nodes is deployed in the campus. The purpose of such testbed is to monitor the indoor environmental condition as well as the network condition. Every sensor node will sample environment data, such as temperature, light, humidity, in a constant sampling rate. We also collect network condition information to do other research programs.

To evaluate the performance of SA-MAC, we will test the energy consumption of sensor nodes in the network, packet loss rate, in different working patterns. In addition, we collect the working pattern of every sensor nodes, and check the stability of sensor nodes in the system. And we also compare the performance between SA-MAC and ContikiMAC.

4.3.1 Energy Consumption

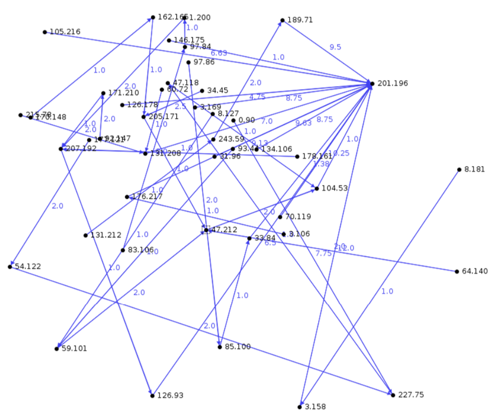
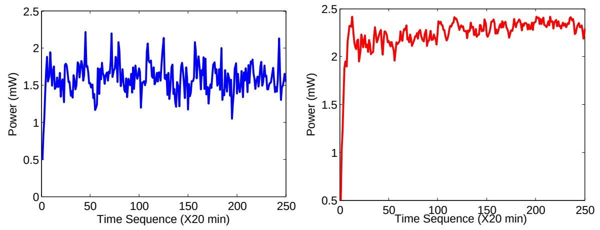


Fig. 11. Network topology

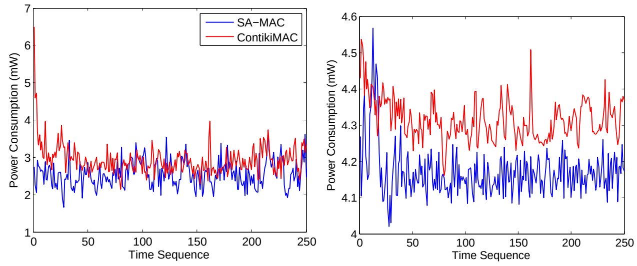


(a) Power consumption for one single node. (b) Average power consumption for nodes in the network.

Fig. 12. Power Consumption in network

In this evaluation, the energy is evaluated through a program called PowerTrace [15], which is a network-level power profiling of low-power wireless systems. The main advantage of this technique is that it could track the power state precisely, and break the energy consumption down into individual activities.

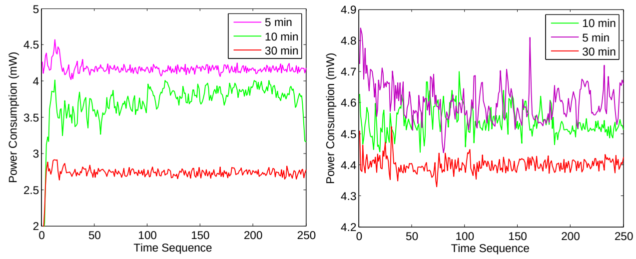
In the first experiment, we setup a wireless sensor network consisting of 45 sensor nodes (including sink node). The sampling rate is one sampled packet every 20 minutes. The network topology is shown in Figure 11. We take one node as example, Figure 12(a) presents the energy consumption during a constant time slot. It is obvious that the average power consumption is less than $2mW$. Obviously, the energy consumption is determined by many factors, such as duty cycle, sampling rate, topology, and so on. However, the key factor is the duty cycle. Initially, we set the wake-up frequency of $8Hz$, which indicates the waking-up interval is $125ms$. The average power consumption for all the sensor nodes in the network is also less than $2.5mW$, the results are presented in Figure 12(b).



(a) Minimum power consumption for one node in network (b) Average power consumption for both SA-MAC and ContikiMAC.

Fig. 13. Power Consumption in two protocols

We also compare the SA-MAC with ContikiMAC, in Figure 13. The first figure plots the power consumption of some particular sensor node in the network, with the minimum power consumption. These two sensor nodes in the network perform as leaf node, which do not have to relay others' data. The second figure, both curves indicate the average power consumption of all the sensor nods in the network, where the red curve is for ContikiMAC, and the blue curve for SA-MAC. The average power consumption in ContikiMAC is around $4.3mW$, while that of SA-MAC is about $4.15mW$. The energy consumption in SA-MAC is lower than that of ContikiMAC.

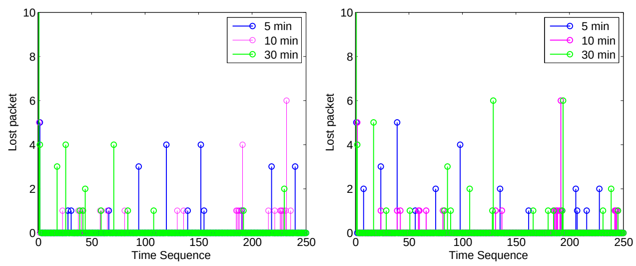


(a) SA-MAC (b) ContikiMAC

Fig. 14. Power Consumption in two protocols

In the third experiment, we compare the energy consumption under the condition of different sampling rates. According to the previous description, in some extent, large sampling rate will lead to relatively high packet loss, and power consumption. We set three different cases to evaluate the performance, the time intervals are set to be $5min$, $10min$, and $30min$, as shown in Figure 14. Obviously, with the sampling rate increases, the average power consumption rise simultaneously. In the real scenario, our system running under the sample rate of every 30 minutes, the energy consumption is acceptable.

4.3.2 Packet Reception Rate



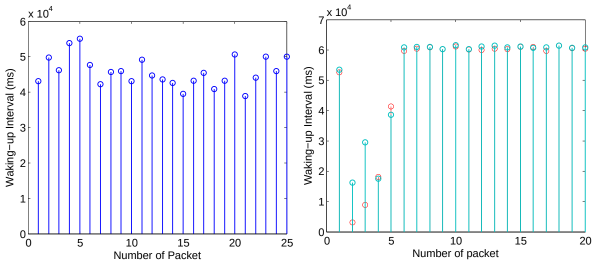
(a) SA-MAC (b) ContikiMAC

Fig. 15. Packet Loss in Evaluation

Packet reception rate is another metric to evaluate the performance of network. In this experiment, we allocate a sequence number to every packet when the time of sampling expires. Through the sequence, we could easily estimate the packet reception rate during the deployment. Figure 15(a) depicts the packet loss condition during the experiment. The packet loss condition is evaluated in three different cases, with the sampling rate in every 5 minutes, 10 minutes, and 30 minutes respectively. In SA-MAC, according to the test, the number of lost packet is under acceptable condition, the packet reception ratio are all above 98% in three different cases. Generally, for long term experiment, the sink receives 10652 packets totally, but lost 150, which leads to roughly 98.6% in packet reception rate.

When it comes to ContikiMAC, Figure 15(b) plots the packet loss condition with the time sequence increase. The packet reception ratio for ContikiMAC in the same three cases are similar, above 97%. Both SA-MAC and ContikiMAC provide high packet reception ratio under real networks.

4.3.3 Working Pattern



(a) One Sender One Receiver.

(b) Multiple Sender One Receiver.

Fig. 16. Waking-up Pattern

This part will present the experiment of the working pattern of sensor nodes, to see if the network will evolve as a TDMA using phases-lock. We test in two different cases: 1) One Sender One Receiver, and 2) Multiple Sender One Receiver. In this experiment, the sampling rate is set to be 10 minutes. As shown in Figure 16(a), after the adjustment from the beginning, the sensor finds its phase and start sending packet in a relatively constance time interval, about 45000 ms. When it comes to the second case, we set ten senders send packet directly to the sink node, and after fluctuation, each node finds their own phases, and the waking-up interval of each nodes become stable. As shown in Figure 16(b), both two different sensors, green and red endure a five times' fluctuation before entering stable condition. We also calculate the correlation coefficients of each packet interval of received packet, and the result is 0.9737, which indicates that the SA-MAC will be stable automatically after an adjusting period. In the first case, the sender finds its phase after three packet transmissions, while in second case it stabilizes after five packet transmissions. In addition, the receiver will finds its phase in a constance period, which presents the SA-MAC works in a stable TDMA condition.

5 Conclusion

In this paper, we present SA-MAC, a sender-initiated low power duty cycling MAC protocol, with self-stabilizing adaptive mechanism. Our protocol improves upon Contiki-MAC, and integrates an transmission and packet detection pattern adaptive to the current network condition. We conduct rigorous analysis on the energy consumption, and other performances based on practical models. We also implement the SA-MAC and test its performance in a sensor networking system. Our evaluation results show that the energy consumption of SA-MAC is lower than other existing low-power duty cycling MAC protocols, and performs in a TDMA like pattern.

References

[1] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the*

2nd international conference on Embedded networked sensor systems, pages 95–107. ACM, 2004.

[2] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1567–1576. IEEE, 2002.

[3] T. Van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 171–180. ACM, 2003.

[4] M. Buettner, G.V. Yee, E. Anderson, and R. Han. 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*, pages 307–320. ACM, 2006.

[5] Adam Dunkels. The ContikiMAC Radio Duty Cycling Protocol. Technical Report T2011:13, Swedish Institute of Computer Science, December 2011.

[6] K. Pister and L. Doherty. Tsmp: Time synchronized mesh protocol. *IASTED Distributed Sensor Networks*, pages 391–398, 2008.

[7] S. Du, A.K. Saha, and D.B. Johnson. R-mac: A routing-enhanced duty-cycle mac protocol for wireless sensor networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 1478–1486. Ieee, 2007.

[8] Y. Sun, S. Du, O. Gurewitz, and D.B. Johnson. Dw-mac: a low latency, energy efficient demand-wakeup mac protocol for wireless sensor networks. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 53–62. ACM, 2008.

[9] A. El-Hoiydi and J.D. Decotignie. Wisemac: An ultra low power mac protocol for multi-hop wireless sensor networks. *Algorithmic Aspects of Wireless Sensor Networks*, pages 18–31, 2004.

[10] 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 *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 1–14. ACM, 2008.