Adaptive Energy-Saving Mechanism for SMAC Protocol in Wireless Sensor Network

Zhou Jieying(✉), Peng Shi, Liu Yinglin, and Huang Shaopeng

School of Data and Computer Science, Sun Yat-sen University,
Guangzhou 510006, Guangdong, China
isszjy@mail.sysu.edu.cn

Abstract. This paper proposes an Adaptive Energy-Saving mechanism to improve the performance of SMAC protocol (AES-SMAC). AES-SMAC is composed of two mechanisms, adaptive synchronous period (ASP) and dual element adaptive contention window (DEACW). In ASP, the cache queue of SMAC is simplified into a limited capacity of M/M/1 queuing model to calculate the node's service efficiency. ASP also introduced a parameter called sleep conflict rate to evaluate the performance of node synchronization. Then ASP adjusts the length of synchronous period reasonably to achieve better performance. AES-SMAC protocol was compared with SMAC, ASP and DEACW via the simulation software NS2 in a scenario of Multi-hop topology. Simulation results show that AES-SMAC achieved better performance with higher throughput, lower average delay and higher energy utilization efficiency.

Keywords: Wireless sensor network · SMAC protocol · Queuing theory · Synchronous period · Contention window

1 Introduction

In the development of wireless sensor network (WSN), MAC (Media Access Control) which provides reliable communication links has been a hot research topic. There are many MAC protocols that have been developed, such as the time division multiple access (TDMA), code division multiple access (CDMA), and contention-based protocols like IEEE 802.11 [1]. Because of the special environment of WSN application, traditional MAC protocol which is based on minimum delay and maximum throughput is not applicable for WSN [2, 3]. WSN focuses on reducing energy consumption while a certain throughput is guaranteed. Based on the characteristic that WSN has limited energy and unique channel access technology, SMAC (sensor-MAC) with dormancy mechanism and based on IEEE802.11 is proposed [4–6]. It aims to reduce energy consumption, enhance scalability of the network and adapt to the dynamic topology changes that it would be suitable for wireless sensor networks. The disadvantages of SMAC are that the throughput, delay and fairness are not considered in this mechanism and fixed synchronous period and contention window cannot adapt to the dynamic changes of the network environment [7, 8]. Therefore an improved protocol–AES-SMAC is introduced

Q. Chen et al. (Eds.): ChinaCom 2016, Part I, LNICST 209, pp. 47–57, 2018.
DOI: 10.1007/978-3-319-66625-9_5

in this paper to solve the above problems. The rest of this paper is organized as follows: the improved SMAC protocol with adaptive energy saving mechanism (AES-SMAC) is proposed in Sect. 2. Section 3 introduces the simulation of AES-SMAC, DEACW, ASP and SMAC via the software NS2 in a scenario of Multi-hop topology, the simulation results of the four mechanisms are compared and analyzed in this section. Sect. 4 draws a conclusion of this paper.

2 Adaptive Energy-Saving Mechanism for SMAC

To overcome the shortcomings of SMAC, this paper proposes an adaptive energy saving mechanism–AES-SMAC to improve the performance of the WSN network which adjusts the synchronous period and the contention window adaptively. AES-SMAC is composed of adaptive synchronous period (ASP) and dual element adaptive contention window (DEACW). In ASP mechanism, to reduce the extra energy consumption resulting from packet re-transmission for packet loss during nodes sleep period, the current network state and synchronization state between nodes are measured based on queuing theory [9] and sleep conflict rate which are used to adjust synchronous period reasonably. While DEACW measures the network channel occupancy and fairness with conflict free ratio and node channel occupancy which are used to adjust contention window adaptively. AES-SMAC improves overall performance of WSN network and reduces energy consumption by combining the advantages of ASP and DEACW mechanisms.

2.1 Introduction of ASP

To overcome the deficiency in the synchronization mechanism of SMAC protocol, this paper proposes the ASP mechanism which measures the network and node synchronization state and adjusts the synchronous period adaptively.

In order to measure the state of the network and adjust the synchronous period adaptively, data packet service rate is used in this paper to measure the traffic flow of the network after a node performs synchronization. It is based on queuing theory and reflects the congestion of the network. In order to reflect the synchronization state between nodes, this paper introduces a parameter col_s to represent the conflict ratio of data packets caused by sleeping nodes. At the same time, a new symbol is added in the control packet to monitor the synchronous time among nodes in the switching process of controlling packet. Once time difference is found to be too large, it immediately performs synchronization at the next sleep/listen cycle. The improved mechanism will calculate the average μ and col_s at the end of a synchronous period. Two values are used to analyze the network congestion and synchronization state, so as to adjust the synchronous period reasonably. ASP mechanism increases the network throughput, reduces the average end-to-end delay and also reduces energy consumption [10], therefore the performance of the network is improved.

(1) Calculation of μ and col_s

First, the service rate μ is defined as

$$\mu = \lambda\left(\sqrt{\left(\frac{1}{4} + \frac{1}{L_q}\right)} + \frac{1}{2}\right) \tag{1}$$

where λ refers to the arrival rate of data packets, L_q refers to number of packets in buffer. It can be seen from formula (1) that the value of μ is proportional to the value of the input rate λ, is inversely proportional to the length of the packet in the queue. In SMAC protocol, if the current network is congested and the clock difference with the next hop is large, the current node can not send out packet. It has to wait for a long time for synchronization, which leads to the increasing of data packets waiting in the buffer queue. Even if the clock drift is small, it still can not complete data transmission when network congestion occur. The value of μ will decrease which reflects that current network load is heavy. When the network load is relatively light, if the input rate of the data packet is great, there may be two kinds of situation occur. In the first situation, synchronization between nodes is very good, data packet in the queue is almost zero, therefore, the value μ will be very large, reflecting that processing speed of the node is fast. In the second situation, the node synchronization is relatively bad, the data packets are stuck in conflict, the the value of μ will decrease, reflecting that processing speed of the node is slow.

Sleep conflict ratio is defined as the ratio of the sleeping conflict to the total conflict in the synchronous period, i.e.

$$\text{col\_s} = \frac{col_{sleep}}{\left(col_{sleep} + col_{other}\right)} \tag{2}$$

where col_{sleep} is the length of conflict caused by sleep, col_{other} is the length of conflict caused by other reasons.

From the definition of col_s we know that, when col_s is large, the value of col_{sleep} is too large in comparison with the col_{other}. This situation occurs because the clock difference of the nodes in the network increases along with time passes by. And the long synchronous period is unable to prevent the occurrence of this situation, resulting in the packet loss caused by node sleep, at this time it is necessary to reduce the synchronous period. And when col_s is small, there are two situations, one is the value of col_{other} is too large in comparison with col_{sleep}, which means that great conflicts are caused by the network congestion although the node synchronous state is good. At this time it is necessary to increase the synchronous period. The other situation is that the network load is very light, the probability of col_{sleep} and col_{other} is relatively small.

In order to measure the current state of the network reasonably, we should take two parameters into consideration when deciding how to adjust the synchronous period. Because col_s is based on the average value of a synchronous cycle, while the value of μ depends on the average number of packets in the buffer queue and the average rate of entry. Therefore it is necessary to record the two values in each sleep /listen cycle. The calculation is as follows: Calculation of service rate μ:

$$\mathrm{E}(L_{\mathrm{q}}) = \frac{\left(L_{\mathrm{q}}^{(1)} + L_{\mathrm{q}}^{(2)} + \cdots L_{\mathrm{q}}^{(syncperiod)}\right)}{syncperiod} \tag{3}$$

$$\lambda = \frac{\left(\mathrm{p}^{(1)} + \mathrm{p}^{(2)} + \cdots \mathrm{p}^{(syncperiod)}\right)}{(syncperiod \times cycletime)} \tag{4}$$

$$\mu = \lambda\left(\sqrt{\left(\frac{1}{4} + \frac{1}{E(\mathrm{q})}\right)} + \frac{1}{2}\right) \tag{5}$$

Calculation of sleep conflict rate col_s:

$$\mathrm{col\_s} = \left(\frac{\left(col_{sleep}^{(1)} + col_{sleep}^{(2)} + \ldots + col_{sleep}^{syncperiod}\right)}{\left(col_{all}^{(1)} + col_{all}^{(2)} + \ldots + col_{all}^{syncperiod}\right)}\right) \tag{6}$$

where

$$\mathrm{col\_all}^{(n)} = \mathrm{col\_sleep}^{(n)} + \mathrm{col\_other}^{(n)} \tag{7}$$

where synchronous period is the number of listening cycle in a synchronous period, cycletime is a monitoring cycle, p is the number of received packages in a listening period.

(2) **Adjustment of synchronous period**

When a synchronous cycle is over, the node will calculate the average service rate and the average sleep conflict ratio in a cycle, and then adaptively adjusts the synchronous period according to the following mechanism, as shown in Algorithm 1:

The parameters of θ, ω, β, α, γ and δ are used to balance the jitters of the col_s and μ where SYNC_MIN and SYNC_MAX are the maximum and minimum values of a synchronous cycle.

In Algorithm 1, the code calculates the value of col_s at first. There are two situations, if col_sleep and col_other are not zero at the same time, then it will calculate the value of col_s. When two value are zero, there is no data exchange in the network or the load of network is light, then col_s is -1. The algorithm adjusts the synchronous period reasonably by analyzing the current network state and node synchronization. So as to achieve the goal of increasing the throughput, reducing delay and improving energy efficiency.

After calculating the value of col_s, if the average number of packets in the buffer area is more than 45 (the queue length of simulation setting is 50), the network is considered to be extremely congested regardless of the value of μ. Considering the relationship among the synchronous period, time delay and the throughput, at this time, it is necessary to reduce the probability of the occurrence of the clock difference.

Therefore, the synchronous period should be appropriately reduced. When average number of data packets waiting in the buffer area is in the range of 10 to 45, the network is considered to be in good state. If $\mu \geq \beta + \omega$, it means that the network state is very good. If col_s $< \delta + \theta$ at the same time, the probability of the increasing clock difference is small. So we can appropriately increase the synchronous period and reduce the energy caused by sending SYNC packets. When col_s $> \gamma + \theta$, we should reduce the synchronous period and keep synchronization between nodes. When $\alpha \leq \mu < \beta$, network load and synchronization are both in balanced state, therefore overall performance of the network will be better without changing synchronous cycle. When $\mu < \alpha$, the network load is heavy. If col_s $< \delta + \theta$ at the same time, then synchronization between nodes is very good. So we can increase the synchronous period appropriately to reduce the occurrence of more conflicts caused by sending SYNC packets. At last, when the average number of packets in the buffer area is less than 10, the load of the network is light. When col_s $> \gamma + \theta$, in order to reduce the packet loss caused by sleep, we should reduce the synchronous period to improve the throughput. If col_s $= -1$ at the same time, then the network load is considered to be extremely light. In order to reduce the energy consumption of nodes, we should increase the period of the synchronization.

```
Algorithm 1 ASP
if ($col_{other}$ !=0) then
    col_s=$col_{sleep}$/($col_{sleep}$+$col_{other}$);
else
    if (col_sleep==0) col_s=-1;
    else col_s=$col_{sleep}$/($col_{sleep}$+$col_{other}$);
end if
if ($L_q$≥ 45) then
    if(col_s≥γ+θ)
    syncperiod=max(syncperiod-2,SYNC_MIN);
else if ( $L_q$< 45&& $L_q$ ≥ 10 ) then
    μ=(sqrt(1/$L_q$+ 0.25)+0.5)*suc/cycleTime_/count;
    if (μ ≥ β+ ω) then
          if(col_s<δ+θ)
          syncperiod=min((syncperiod+1),SYNC_MAX);
          if(col_s>γ+θ)
          syncperiod=max((syncperiod-1),SYNC_MIN);
    else if ( col_s < δ+θ && μ < α+ω) then
          syncperiod=min((syncperiod+1), SYNC_MAX);
    end if
else
    if ( col_s > γ+θ )
    syncperiod=max((syncperiod-2),SYNC_MIN);
    else if(col_s=-1)
    syncperiod=min((syncperiod+2),SYNC_MAX);
end if
```

The basic algorithm is described above. From the analysis, it can be seen that, in order to calculate the sleep conflict ratio and service rate at the end of the period, the node must record the length of a cycle and the number of packets in the buffer area. And then, we can adjust the length of the synchronous cycle and improve the performance of overall network.

(3) **Synchronous monitoring of the node**

For the above algorithm, there are some situations that are not considered. If the synchronous period fluctuates frequently and dramatically, the performance of the whole network becomes worse. So the synchronous monitoring mechanism is proposed in RTS/CTS exchange to monitor clock drift between nodes. If the clock drift exceeds a threshold value, it directly sends SYNC packages in the next waking-up without waiting for the end of the cycle. Specific ideas are as follows (Fig. 1):

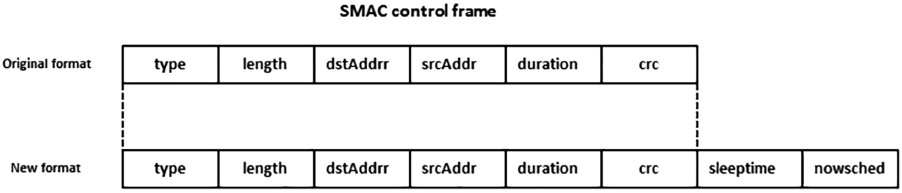

Fig. 1. SMAC control frame modification

First, two values are inserted in the RTS control frame. The monitoring mechanism needs to add two values in the control frame. Sleeptime is used to record the time left for the node to sleep, nowsched records the current scheduling table of this node. After the node receives RTS control frame, according to the two values, it can be judged whether two nodes are in the same scheduling table and clock difference can be calculated. If it is larger than the threshold value, then the node sends a synchronizing packet at the next waking-up.

Therefore nodes may encounter the following situations, as shown in Fig. 2. The receiving node sends the SYNC packet to achieve synchronization in the next waking up. We don't use the source node to send SYNC packet because the transmission may be start when the receiving node still sleeps. The synchronization is very good without the need to send SYNC packet. The threshold value will be set larger than before the sudden SYNC broadcast may affect data transmission of other nodes.

3 Simulation Results and Performance Analysis

In this section, AES-SMAC protocol is compared with three other schemes: SMAC, ASP and DEACW. Multi-hop topology is used to simulate the four schemes in this paper. CBR model is used to generate the data stream in the network. Sending interval is used to simulate the load of the network. The smaller the interval is, the more data is transmitted in unit time, then load of the network is more heavier. For each of these

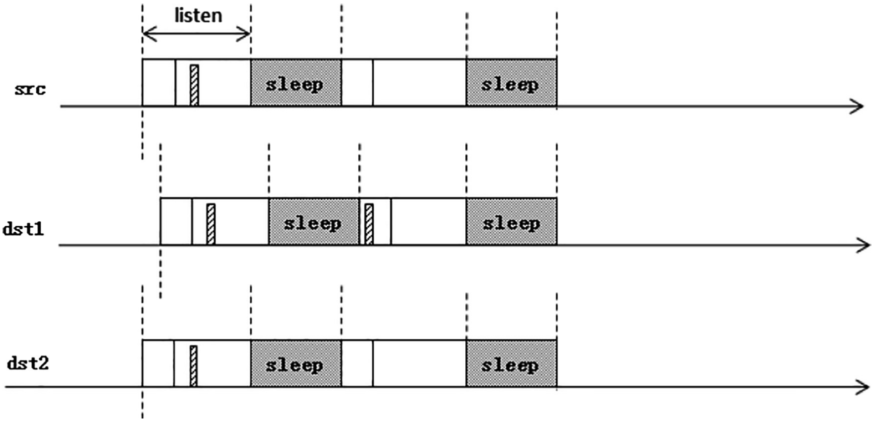

Fig. 2. Node synchronization

schemes, the simulation is carried out separately in each scenario. Results of these protocols are analyzed and compared after the simulation. In WSN, data usually needs to be transmitted to the sink node through multi-hop. In this paper, this situation is simulated by the topology of multi hop in Fig. 3. The topology is composed of 10 nodes. The distance between two nodes is 200 m. The simulation time is 1000 s. The CBR data stream starts at 50 s and ends at 800 s.

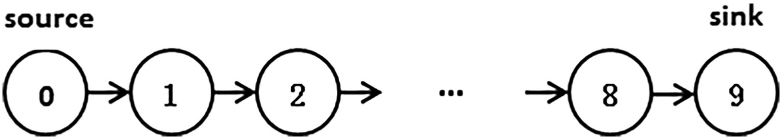

Fig. 3. Multi-hop transmission topology

In this simulation, the transmission range of each node is 250 m. The distance between nodes is determined according to the setting of the scene. And CBR (Constant Bit Rate) is used as the generator of the data stream in this simulation. Packet size of the data stream is 256 bytes. UDP protocol is used as transport layer protocol.

Figure 4 shows that in the multi-hop topology, the through put of SMAC, DEACW, ASP, and AES-SMAC changes with the increase of sending interval. For AES-SMAC, in heavy load area, the node adjusts the contention window adaptively and reduces the probability that multiple nodes simultaneously compete the channel. At the same time, AES-SMAC monitors network traffic situation through ASP mechanism, increases synchronous period and reduces the transmission of controlling packet. So the throughput of AES-SMAC is better than ASP and DEACW in heavy load area.

Figure 5 shows that in the multi-hop topology, the delay of four mechanisms changes with the increase of sending interval. In the heavy load area, the network delay

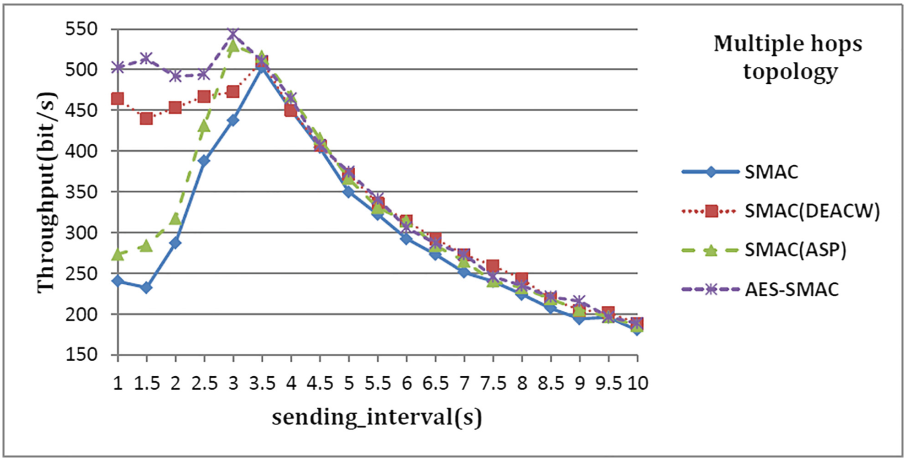

Fig. 4. Throughput variation under different sending interval

is mainly caused by conflicts, exposed and hidden terminal problems. In ASP, conflicts increase easily when the exposed and hidden terminal problems occur, because it adopts the mechanism of fixed contention window, so the improvement is not great. In light load area, ASP which focuses on synchronization has smaller delay. In the heavy

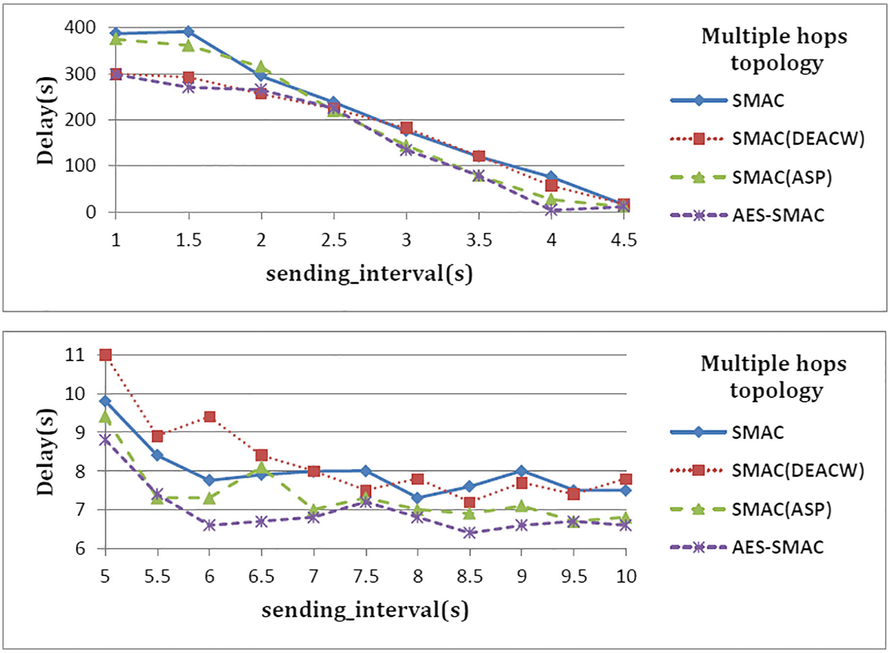

Fig. 5. Delay variation under different sending interval

load area, because a large amount of conflicts and SYNC packet transmission are reduced, the delay of AES-SMAC is significantly better than SMAC and ASP. While in light load area, because the ASP mechanism reduces the impact of sleeping delay, the delay of AES-SMAC is better than the other 3 mechanisms.

Figure 6 shows that in the multi-hop topology, the energy consumption of the four mechanisms changes when sending interval increases. Since throughput and SYNC packet transmission are increased in the adjusting process, the overall energy consumption of ASP is larger than DEACW. In the heavy load area, because the increase of throughput is great, the energy consumption of DEACW is larger than ASP.

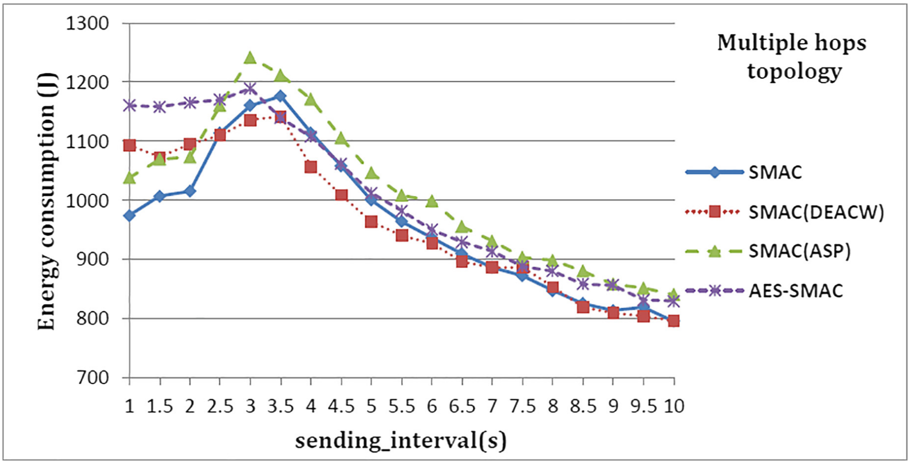

Fig. 6. Energy consumption variation under different sending interval

In light load area, energy consumption of AES-SMAC is between DEACW and ASP. Because DEACW reduces energy consumption caused by the idle listening time, while the ASP increases the energy consumption because of sending control frame. The energy required for transmission is much more than idle listening, so the energy consumption of AES-SMAC is slightly more than that of SMAC in light load area.

Figure 7 shows that in the multi-hop topology, the energy efficiency of the four mechanisms changes when sending interval increases. In heavy load area, AES-SMAC reduces conflicts of data packets by adjusting the contention window according to the network state, meanwhile it reduces the transmission of SYNC packets by slightly increasing the synchronous period according to the network load, therefore AES-SMAC reduces the energy consumption from two aspects so energy efficiency is very good. In light load area, idle listening is the main reason of energy consumption. Therefore, AES-SMAC needs to consider the energy consumption caused by idle listening. But because ASP mechanism increases the synchronous period in light load area, the energy efficiency of AES-SMAC is better than DEACW in light load area.

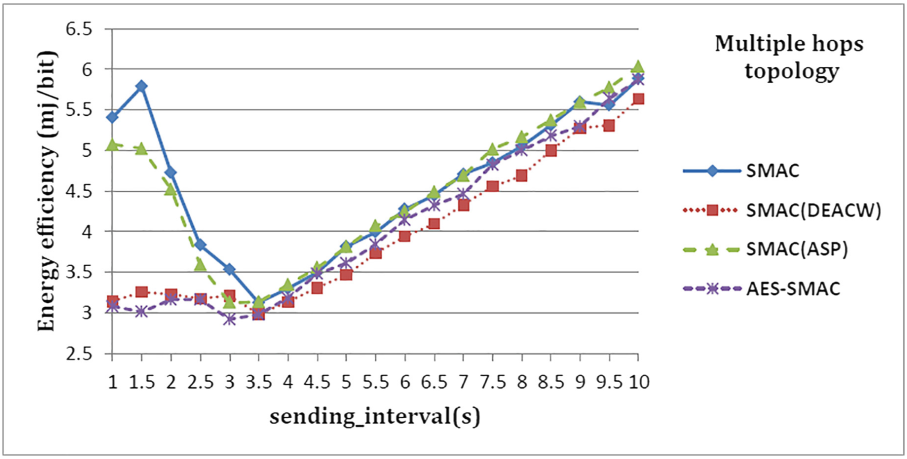

Fig. 7. Energy efficiency variation under different sending interval

4 Conclusion

This paper proposed an adaptive energy-saving mechanism to overcome the shortcomings of SMAC protocol. The AES-SMAC protocol is composed by ASP and DEACW. In ASP, the synchronous period is adjusted adaptively according to the network congestion states and synchronization situation between nodes. Meanwhile the network load and fairness are used in DEACW to adjust contention window reasonably.

The comparison experiments of AES-SMAC protocol with other relative protocols are carried out via simulation software NS2. Different network topologies are used in the experiment. Simulation results show that AES-SMAC protocol achieves higher throughput, lower average delay and higher energy utilization efficiency than SMAC and other relative protocols, therefore the performance of the network is improved efficiently.

Acknowledgement. This paper is financially supported by the program of Guangdong Science and Technology (Grant No. 2015A010103007) and the program of Guangzhou Science and Technology (Grant No. 2015A010103007).

References

1. LAN MAN Standards Committee of the IEEE Computer Society: Wireless LAN medium access control (MAC) and physical layer (PHY) specification, IEEE Std 802.11-1997 edition. IEEE, New York (1997)
2. Yang, O., Heinzelman, W.B.: Modeling and performance analysis for duty-cycled MAC protocols with applications to S-MAC and X-MAC. IEEE Trans. Mob. Comput. **11**(6), 905–921 (2012)

3. Feng, H., Ma, L., Leng, S.: A low overhead wireless sensor networks MAC protocol. In: 2010 2nd International Conference on Computer Engineering and Technology (ICCET) (2010)
4. Estrin, D., Ye, W., Heidemann, J.: An energy-efficient mac protocol for wireless sensor networks. In: Proceedings of Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 1567–1576. IEEE (2002)
5. Heidemann, J., Ye, W.: Medium access control in wireless sensor networks. In: Raghavendra, C.S., Sivalingam, K.M., Znati, T. (eds.) Wireless Sensor Networks, pp. 73–91. Springer, Boston (2004). doi:10.1007/978-1-4020-7884-2_4
6. Estrin, D., Ye, W., Heidemann, J.: Medium access control with coordinated adaptive sleeping for wireless sensor networks. IEEE ACM Trans. Netw. **12**(3), 493–506 (2004)
7. Yang, O., Heinzelman, W.: Modeling and throughput analysis for SMAC with a finite queue capacity. In: 2009 5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP) (2009)
8. Heinzelman, W., Ray, S., Demirkol, I.: ADV-MAC: advertisement-based MAC protocol for wireless sensor networks. In: 5th International Conference on Mobile Ad-hoc and Sensor Networks, MSN 2009 (2009)
9. Chouhan, L., Trivedi, A.: Priority based MAC scheme for cognitive radio network: a queuing theory modelling. In: 2012 Ninth International Conference on Wireless and Optical Communications Networks (WOCN) (2012)
10. Pottie, G.J., Kaiser, W.J.: Embedding the internet: wireless integrated network sensors. Commun. ACM. **43**(5), 51–58 (2000)