Article ID：0253-2778(2010)10-1054-11

# HMAC：An energy efficient MAC protocol for wireless sensor networks

SU Yu，QU Yugui，LIN Zhiting

(*Department of Electronic Engineering and Information Science，University of Science and Technology of China，Hefei* 230027，*China*)

**Abstract**：Duty-cycle MAC protocols widely used in wireless sensor networks lead to extra end-to-end delivery latency，while the existing solutions，such as RMAC (the Routing enhanced MAC protocol)，are only applicable to light-load traffic. A new MAC protocol called HMAC is proposed，which avoids the unnecessary energy consumption caused by loss events without sacrificing the performance of packet delivery latency. By using two scheduling frames，HMAC realizes efficient multihop packet delivery in a single cycle，and ensures that bad link conditions do not impact the downstream nodes. Compared with S-MAC and RMAC，it is shown that HMAC outperforms these protocols in a heavy-load traffic scenario，with higher energy efficiency and lower delivery latency.

**Key words**：wireless sensor networks；medium access control；heavy traffic；loss event

**CLC number**：TP393 **Document code**：A doi：10.3969/j.issn.0253-2778.2010.10.010

# 一种高效节能的无线传感器 MAC 协议

苏　宇，屈玉贵，蔺智挺

（中国科学技术大学电子工程与信息科学系，安徽合肥 230027）

**摘要**：无线传感器网络基于睡眠调度机制的 MAC 协议存在传输延迟较大的缺点. 针对现有的解决方案多只适用于低负载环境的局限性，提出了一种改进方案— HMAC. 通过使用两个调度帧，HMAC 可在一个周期内实现高效多跳传输，同时保证恶劣链路条件不会影响下游节点. 理论分析和仿真实验表明，与 S-MAC 和 RMAC 相比，采用该协议可以在重负载下有效提高能量利用率，并降低传输延迟.

**关键词**：无线传感器网络；媒介访问控制；重负载；丢包

**Received**：2009-08-28；**Revised**：2009-11-11

**Foundation item**：Supported by National Natural Science Foundation of China (60872009，60602016)，Hi-Tech Research and Development Program (863) of China (2007AA01Z428，2009AA01Z148) and Natural Science Foundation of Anhui (ZD2008005-2，ZD200904，JK2009A013，JK2009A025).

**Biography**：SU Yu，female，born in 1985，master. Research field：wireless sensor networks. E-mail：elv5tion@mail.ustc.edu.cn

**Corresponding author**：QU Yugui，PhD/Prof. E-mail：ygqu@ustc.edu.cn

## 0 Introduction

Wireless sensor networks have significant potential in applications such as target tracking and monitoring environmental phenomena. In these applications, sensor nodes are usually battery-powered and left unattended after deployment. The limited battery capacity severely limits the network lifetime. In a resource-constrained system, for example, WSNs, it is an important issue to keep energy efficient.

Studies show that significant power is wasted in idle listening, that is, listening to an idle channel and waiting to receive packets even though there is no traffic in the network. Many works based on duty cycling (e. g. , Ref. [1-3]) have been proposed to mitigate the energy consumption of idle listening. The basic idea of these mechanisms is that sensor nodes periodically switch between listening mode and sleeping mode.

An overview of S-MAC[1] is shown in Fig. 1. S-MAC uses the RTS/CTS scheme from IEEE 802. 11 to reduce the collisions caused by the hidden-node problem. Sensor nodes in S-MAC follow a periodic listen/sleep schedule. The listening period is further divided into a Sync period and a Data period. During the Sync period, an independent protocol is used to synchronize the clocks of all the sensor nodes. The Data period is used to deliver data packet. All sensor nodes should go back to sleep at the beginning of the Sleep period except those that are communicating. The sender and the receiver can return to the sleeping mode only if the packet delivery is successfully completed. S-MAC keeps energy efficient since it forces sensor nodes to operate at a low duty cycle by putting them into periodic sleeping instead of idle listening.

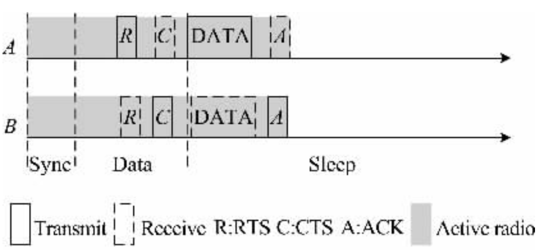


**Fig. 1 S-MAC overview**

Although duty-cycle MAC protocols save more energy than standard MAC protocols, they have been proved to be limited in end-to-end delivery latency. Generally, if a node wants to transmit a packet to a neighbor node, it has to wait until the node wakes up to receive the packet. These protocols cause high latency in multihop packet delivery, since a data packet can be forwarded over only one hop in each cycle.

Various methods have been proposed for overcoming the limitation of duty cycling. RMAC[4] exploits the cross-layer routing information to avoid significant end-to-end delivery latency while keeping energy efficientcy. As shown in Fig. 2, RMAC employs series of control frames, called PION frames (pioneer control frames), to realize efficient multihop packet delivery within a cycle. The PION frame is used to request communication with the downstream node and confirm receipt of the PION from the upstream node. By forwarding the PION frames through the stream during the Data period, the transmission schedule can be set before data packets arrive. Therefore, a data packet can be efficiently forwarded over multihops in the Sleep period.

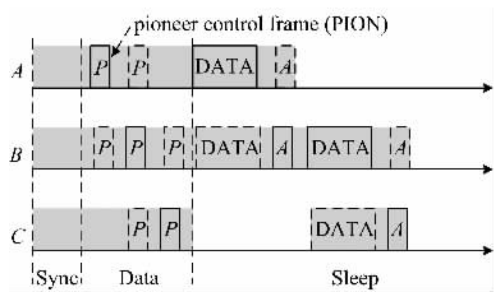


**Fig. 2 RMAC overview**

References show that RMAC has a great performance in a light-load traffic scenario. However, it performs quite poorly when the traffic is busy. As the traffic load increases, the packet loss rate, including the losses of data, PION and ACK frames, rises substantially. Loss events have a tremendous negative impact on RMAC, because the downstream nodes suffer from extra energy consumption caused by these loss events. Specifically, if a PION is lost, the upstream node cannot receive the confirmation of receipt for the PION frame from the downstream node. This upstream node will not forward the data packet to the downstream node in the current cycle, but it will try again in the next cycle with a fresh PION. However, the downstream node does not know the above facts. It wakes up unnecessarily, waits to receive the data packet, but gets nothing until timeout. Fig. 3 shows the worst case. A PION frame successfully goes through several hops along downstream, but the source node loses the acknowledgement PION frame from the downstream node. In this case, all nodes along downstream wake up, wait, and go back to sleep with nothing after timeout. Similarly, if a data or an ACK packet is dropped, no retry occurs in the current cycle. Nodes along downstream also have to bear the unnecessary wakeups and listening.

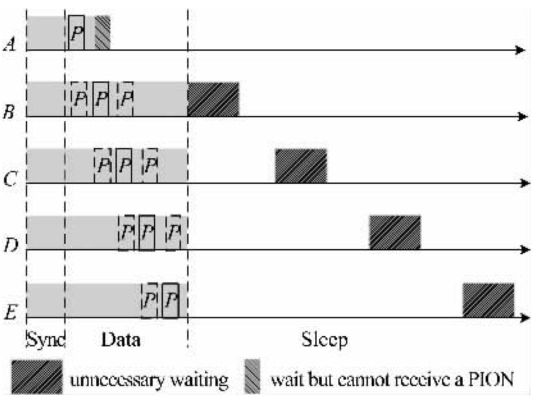


**Fig. 3 The worst case of PION loss events**

Motivated by the above problems, in this paper, we present HMAC, which introduces two scheduling frames that allow efficient multihop packet delivery per cycle in a heavy-load traffic scenario. On the one hand, the scheduling frames are used to set the wakeup time and transmission schedule for sensor nodes, thus HMAC can forward a data packet through multiple hops in an operational cycle. On the other hand, by using the scheduling frames, HMAC ensures that loss events will not disturb unrelated downstream nodes, namely, the downstream nodes will not unnecessarily wait to receive packets when a loss event happens. Therefore, HMAC avoids the extra energy consumption caused by loss events, which happens frequently when the traffic load is heavy.

We organize this paper as follows. The design of HMAC is described in Section 1. Then we analyze its performance of energy consumption in Section 2. In Section 3, we present the simulation results of HMAC, including a comparison with that of S-MAC and RMAC. The related work will be discussed in Section 4. Finally we state our conclusions in Section 5.

## 1 HMAC design

### 1.1 Overview

Fig. 4 shows an overview of the operation of HMAC. Each operational cycle is divided into three periods: Sync, Data and Sleep. Similar to previous work, HMAC uses a separate protocol to synchronize the clocks on sensor nodes during the Sync period. Instead of separating the Data and Sleep period for respective purposes, HMAC integrates them together, which is different from previous mechanisms.

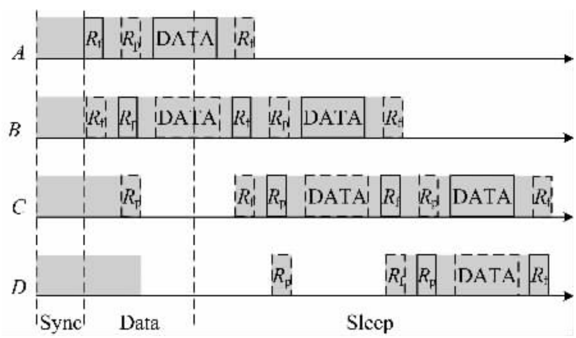


**Fig. 4 HMAC overview**

The basic concept of HMAC is to cut down the unnecessary energy consumption caused by loss events through the use of our proposed scheduling frames, since this kind of consumption cannot be ignored in the heavy-load traffic situation. Even though the wireless network links are unstable, the scheduling frames can increase link utilization and sensor node lifetime, thus improving the stability of the network eventually.

The two scheduling frames, an $R_f$ (formal request frame) and an $R_p$ (pre-request frame), should be successfully transmitted before data packet delivery starts. Both of the frames play a dual role. Generally speaking, an $R_f$ can be regarded as ACK/RTS frames, while an $R_p$ can be regarded as RTS/CTS frames. These frames are used to set the transmission schedule. Each sensor node that has never taken part in packet delivery before should wake up at the scheduled time, and find out whether it is the next hop node. If not, it will check again after a certain period of sleeping. We will discuss $R_f/R_p$ frames and the wakeup schedule in detail in the following subsections.

### 1.2 $R_f/R_p$ frames

RTS/CTS from IEEE 802.11 is replaced by $R_f/R_p$ in HMAC. It makes the relaying of multihop packets efficient and reliable.

As mentioned before, both of the frames have two functions. An $R_f$ frame is used to confirm receipt of the data packet from the upstream node, like an ACK, and to request communication with the downstream node, like an RTS. An $R_p$ frame tells the upstream node that its request has arrived at the downstream node, like a CTS, and asks for communication with the next hop node in advance, like an RTS. Besides all the fields in RTS/CTS, $R_f/R_p$ also contains some cross-layer routing information, such as the address of the final destination and number of hops. We can set the transmission schedule based on the information. This is the first function of $R_f/R_p$ scheme.

Another function of $R_f/R_p$ is to reduce the negative impact of frequent loss events. By using the scheduling frames, HMAC guarantees that the downstream nodes will not waste energy waiting and receiving data packets unless the last-hop data delivery is successful. In particular, if a relay node wants to stay awake for receiving a data packet, it has to figure out whether the last-hop transmission of the data packet succeeds first, and an $R_f$ frame from the upstream node arriving at the relay node before timeout stands for this. As shown in Fig. 4, before the data packet arrives, $C$ has already received an $R_f$ and an $R_p$. They both can be regarded as a request from $B$, but their functions are not exactly the same. The request from the $R_p$ that arrived earlier is just a pre-request, while the request from the later $R_f$ is a formal request. The $R_p$ notifies $C$ that it may be the next hop node, but $C$ will not start waiting to receive data packets unless it gets the $R_f$ before timeout.

### 1.3 Drawbacks of the scheduling frames

The scheduling frames designed for heavy-load traffic can reduce power and latency by providing additional information to downstream nodes in a multihop transmission. Although they can be used in light-load traffic scenarios, it seems quite unnecessary since there are much less packet losses and retransmissions. Besides, relay nodes have to receive more control frames and switch states more frequently, which causes extra energy consumption. If the total energy consumption of the network is small, for example, when the traffic load is low, this extra consumption will take a considerable proportion of the total power, which is unreasonable. Therefore, the scheduling frames have the limitations in the light-load traffic scenario, and they cannot improve the network performance obviously in this condition.

### 1.4 Data transmission

In each operational cycle, packet transmissions start at the beginning of the Data period, and finish at the end of the Sleep period. HMAC forbids data packet transmissions except when the corresponding $R_f/R_p$ frames have been successfully sent.

As illustrated in Fig. 4, $A$ sends an $R_f$ as soon as the Data period starts, asking for a communication with $B$. After receiving the $R_f$, $B$ transmits an $R_p$, which is a confirmation of the receipt of the $R_f$ to $A$ and a pre-request to $C$. If the scheduling frames are successfully delivered, $A$ can relay the data packet to $B$. $B$ stays awake to receive it, while $C$ goes back to sleep in order not to disturb the current communication. After $B$ receives the data packet, it sends an $R_f$, which informs $A$ that the data frame has already arrived, and informs $C$ that it is the next hop node. In other words, the $R_f$ not only signifies a successful data packet transmission in the last hop, but also indicates the starting of packet deliveries in the next hop.

The following packet deliveries operate similarly: the upstream node transmits an $R_f$ first, then its downstream node sends an $R_p$, the data packet is relayed to the downstream node after that. Such a process repeats itself among different sender/receiver pairs along the downstream throughout the Data and Sleep periods, thus a data packet can be forwarded over multihops in a cycle.

Here we define the entire time period to forward a data packet over one hop as $T_{\text{one-hop}}$, which can be expressed as follows:

$$T_{\text{one-hop}} = T_{R_f} + T_{R_p} + T_{\text{data}} + 3\text{SIFS} \tag{1}$$

where $T_{R_f}$, $T_{R_p}$ and $T_{\text{data}}$ indicate the delivery latency of an $R_f$, an $R_p$ and a data frame, respectively.

Besides, we assume that the Data period starts at $T=0$, the current one-hop period is the $i$-th one-hop. As depicted in Fig. 4, node $A$ demands to communicate with its downstream node by sending an $R_f$ during $[0, T_{R_f}]$. The rest of the sensor nodes wonder who the receiver is, so they stay awake and expect the request from $A$. Thus when $i=1$, all nodes should be awake at $T=0$. An $R_p$ is transmitted by $B$ during $[T_{R_f}+\text{SIFS}, T_{R_f}+\text{SIFS}+T_{R_p}]$. After that, all sensor nodes turn to sleeping mode except $A$ and $B$ that are communicating. When $i=2$, $B$ sends an $R_f$ at $T=T_{R_f}+T_{R_p}+T_{\text{data}}+3\text{SIFS}$. $C$, which has got the pre-request from $B$, should wake up at this time to receive the formal request. At the same time, other sensor nodes should wake up at $T=(T_{R_f}+T_{R_p}+T_{\text{data}}+3\text{SIFS})+(T_{R_f}+\text{SIFS})$, to find out if it is the next hop. Similarly, when $i=3$, $D$ wakes up at $T=2(T_{R_f}+T_{R_p}+T_{\text{data}}+3\text{SIFS})$. A sensor node that has never received a request should wake up at $T=2(T_{R_f}+T_{R_p}+T_{\text{data}}+3\text{SIFS})+(T_{R_f}+\text{SIFS})$.

In conclusion, sensor nodes should wake up according to the rules as follows:

(Ⅰ) When $i=1$, for all sensor nodes in the network:

$$T_{\text{wakeup}(i)} = 0 \tag{2}$$

(Ⅱ) When $i \geqslant 1$, for the node which has just received a pre-request during the $(i-1)$-th one-hop-period:

$$T_{\text{wakeup}(i)} = (i-1) \cdot (T_{R_f} + T_{R_p} + T_{\text{data}} + 3\text{SIFS}) \tag{3}$$

(Ⅲ) When $i \geqslant 1$, for sensor nodes that never received requests before:

$$T_{\text{wakeup}(i)} = (i-1) \cdot (T_{R_f} + T_{R_p} + T_{\text{data}} + 3\text{SIFS}) + (T_{R_f} + \text{SIFS}) \tag{4}$$

### 1.5 Retransmission

We make another assumption here that a data packet can go through $n$ hops within each cycle. Generally, if the transmission failure occurs during the $i$-th $(1 \leqslant i \leqslant n-2)$ one-hop-period, the retransmission should start at

$$T_{\text{retrans}} = (i+1) \cdot T_{\text{one-hop}} \tag{5}$$

However, if the failure happens during the $(n-1)$-th or $n$-th one-hop-period, the retransmission has no choice but to wait until the next cycle, since $(n-1)+2$ and the $(n+2)$ are out of the hop range of an operational cycle.

As shown in Fig. 5, the data delivery is frustrated in the second hop. In this case, $C$ will not send an $R_f$. However, other nodes do not know it. $D$ still wakes up at the scheduled time, waits, but cannot receive the formal request $R_f$ until timeout. $D$ has to go back to sleep even though it has a pre-request from $C$. Consequently, downstream nodes that expect the pre-request from $D$ cannot get anything, since $D$ is already asleep. In addition, node $B$ recognizes that it fails to transmit the data packet, because there is no $R_f$ arriving at $B$ until timeout. $B$ will retransmit with

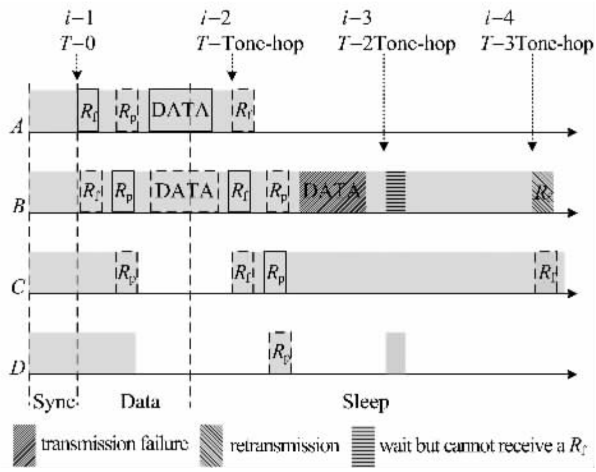


**Fig. 5 An example handling data packet loss using HMAC**

a fresh $R_f$ when the fourth one-hop-period begins, namely, $T=3T_{\text{one-hop}}$.

Similarly, if an $R_f$ or an $R_p$ is dropped, retransmissions are also performed according to Eq. (5) as long as the failure occurs during the first $n-2$ one-hop-periods of each cycle.

## 2 System model

For analysis purposes, a detailed and precise model of a wireless channel is necessary. In this section, we select the Gilbert model[6], a discrete time Markov model, to analyze burst loss.

Fig. 6 gives the outline of the Gilbert model. This model contains two states: a non-loss state and a loss state. We use states 0 and 1 to indicate them respectively. Let $p$ be the probability of transitioning from state 0 to state 1, and $q$ be the probability of transitioning from state 1 to state 0.

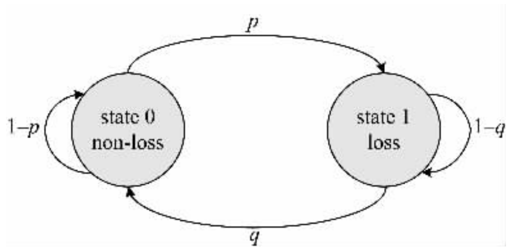


**Fig. 6 The Gilbert model**

We can compute the state probabilities $\pi_0$ for state 0 and $\pi_1$ for state 1:

$$\pi_0 = \frac{q}{p+q},\ \pi_1 = \frac{p}{p+q} \tag{6}$$

As presented in Ref. [7], here we use the Gilbert model to analyze the energy consumption performance of HMAC, including a comparison with that of RMAC.

Let us consider HMAC first. In HMAC, retransmissions can be triggered by the loss of an $R_f$ frame, an $R_p$ frame, a data packet and another $R_f$ frame. The first $R_f$ is sent by the upstream node to initiate a one-hop-period. The last $R_f$ is sent by the downstream node when this period ends. Therefore, the one-hop loss probability $P_{h1}$ can be calculated as

$$(P_{h1})_H = 1-\pi_0^4 \tag{7}$$

The expected one-hop arrival probability is

$$(P_h^r)_H = 1-[(P_{h1})_H]^x = 1-(1-\pi_0^4)^x \tag{8}$$

where $x$ represents the number of transmissions.

In HMAC, all the retransmissions occur in the current cycle except that a failure happens during the last one-hop-period or the one before it. We do not take the two special incidents into account for simplifying our analysis. In other words, the following analysis is carried out under the assumption that retransmissions always occur in the current cycle.

Assuming that the total energy consumption from the transmission failure to the retransmission is $W_5$, $W_6$, $W_7$ and $W_8$, when a requesting $R_f$ frame, an $R_p$ frame, a data packet and an acknowledging $R_f$ frame is dropped, respectively, we can express the average energy consumption in a failed one-hop transmission when a loss event occurs:

$$(E'_h)_H = \pi_1 \cdot (\varepsilon + W_5) + (\pi_0\pi_1)\cdot(2\varepsilon + W_6) + (\pi_0^2\pi_1)\cdot(3\varepsilon + W_7) + (\pi_0^3\pi_1)\cdot(4\varepsilon+W_8) \tag{9}$$

where $\varepsilon$ denotes the average energy consumption for sending a packet.

Since the values of $W_5$, $W_6$, $W_7$ and $W_8$ are quite close, we can rewrite Eq. (9) as

$$(E'_h)_H \approx \pi_1 \cdot (\varepsilon + W_5) + (\pi_0\pi_1)\cdot(2\varepsilon + W_5) + (\pi_0^2\pi_1)\cdot(3\varepsilon + W_5) + (\pi_0^3\pi_1)\cdot(4\varepsilon+W_5) \tag{10}$$

Now we can obtain the average energy consumption for one piece of data through one hop:

$$(E_h^r)_H=[(P_{hl})_H]^x\cdot x\cdot(E'_h)_H+\sum_{i=1}^{x}[(P_{hl})_H]^{i-1}\cdot\pi_0^4[(i-1)\cdot(E'_h)_H+4\varepsilon]\qquad(11)$$

In the first term of Eq. (11), $[(P_{hl})_H]^x$ signifies the event probability that all $x$ transmissions failed through one hop, and $x\cdot(E'_h)_H$ is the corresponding energy consumption. In the last term, $[(P_{hl})_H]^{i-1}\cdot\pi_0^4$ represents the event probability that the transmission through one hop fails for the first $i-1$ times but finally succeeds at the $i$-th, and the corresponding energy consumption is $[(i-1)(E'_h)_H+4\varepsilon]$.

Then, we consider RMAC. Since the loss of a PION, an ACK and a data packet can trigger a retransmission, the one-hop loss probability can be calculated as

$$(P_{hl})_R=1-\pi_0^3\qquad(12)$$

Consequently,

$$(P_h^r)_R=1-[(P_{hl})_R]^x=1-(1-\pi_0^3)^x\qquad(13)$$

$$(E'_h)_R=\pi_1\cdot(\varepsilon+W_1)+(\pi_0\pi_1)\cdot(2\varepsilon+W_2)+(\pi_0^2\pi_1)\cdot(3\varepsilon+W_3)\qquad(14)$$

$$(E_h^r)_R=[(P_{hl})_R]^x\cdot x\cdot(E'_h)_R+\sum_{i=1}^{x}[(P_{hl})_R]^{i-1}\cdot\pi_0^3[(i-1)\cdot(E'_h)_R+3\varepsilon+W_4]\qquad(15)$$

where $W_1$, $W_2$ and $W_3$ represent the total energy consumption of downstream nodes on unnecessarily waiting to receive packets, when a PION, a piece of data and an ACK frame is dropped, respectively. $W_4$ indicates the total energy consumption during the Sync period of the next cycle.

When the traffic load is heavy, $W_1$, $W_2$ and $W_3$ are much larger than $W_5$, $W_6$, $W_7$ and $W_8$. It is because that there are more affected nodes and longer waiting periods that should be regarded when computing the former ones, which causes a waste of energy when unnecessarily waiting to receive packets.

When computing the former ones, multiple nodes along downstream should be considered, and a longer waiting period is needed as the retransmissions of RMAC always occur in the next cycle. At the same time, when we calculate the latter ones, only two sensor nodes that are communicating should be taken into account, and the waiting period is quite short since the retries start within the current cycle. In addition, the more hops a packet can go through per cycle, and the earlier a loss event happens, the greater $W_1$, $W_2$ and $W_3$ will be.

So far, we can prove that:

$$(E_h^r)_R>(E_h^r)_H\qquad(16)$$

Please refer to the Appendix for detailed proof.

The total energy consumption for $M$ packets through $n$ hop is

$$E_e^r=\sum_{i=1}^{n}M(P_h^r)^{i-1}E_h^r\qquad(17)$$

where $M(P_h^r)^{i-1}$ indicates the number of arrival packets after $(i-1)$-th hop.

Therefore, the average energy consumption for a successful arrival packet can be calculated as

$$E_{avg}^r=\frac{E_e^r}{M(P_h^r)^n}=\sum_{i=1}^{n}(P_h^r)^{i-n-1}E_h^r\qquad(18)$$

Since $(E_h^r)_R>(E_h^r)_H$,

$$\frac{(E_{avg}^r)_R}{(E_{avg}^r)_H}=\sum_{i=1}^{n}\left[\frac{1-(1-\pi_0^3)^x}{1-(1-\pi_0^4)^x}\right]^{i-n-1}\cdot\frac{(E_h^r)_R}{(E_h^r)_H}>1\qquad(19)$$

From the above analysis, it can be concluded that HMAC is more energy efficient than RMAC under the conditions of heavy traffic loads. Similarly, we can prove that HMAC outperforms S-MAC at energy consumption under such circumstances.

## 3 Experimental results

### 3.1 Parameters settings

We simulate S-MAC (without adaptive listening), RMAC and HMAC using ns-2 simulator. The purpose is to compare their performances, based on average sensor power and end-to-end delivery latency. The key parameters used in the simulations are listed in Tab. 1. We set the transmission range and the carrier sensing range to 250 m and 550 m, respectively. The simulated time is 1 000 s and the data packet size is 50 bytes. We keep the same duty cycle (10%) for S-MAC, RMAC and HMAC throughout the simulations. Besides, we use the same parameters related to duty cycle when simulating these MAC protocols, such as the time durations of each period.

**Tab. 1　Parameter used in the simulation**

| | | | |
|---|---|---|---|
| RX range | 250 m | SIFS | 5 ms |
| CS range | 550 m | bandwidth | 20 Kbps |
| transmit power | 0.5 W | duty cycle | 10% |
| receive power | 0.5 W | RTS/CTS/ACK | 10 bytes |
| idle power | 0.45 W | PION/$R_f$/$R_p$ | 14 bytes |
| sleep power | 0.05 W | data | 50 bytes |

### 3.2　Overview of scenarios

The simulations are carried out in a 1 000 m×1 000 m network area. The network topology is shown in Fig. 7. There are 50 sensor nodes randomly distributed in the network area, while the only sink node is fixed at (1000, 1000). The network topology ensures that there is at least one route from each sensor node to the sink node.

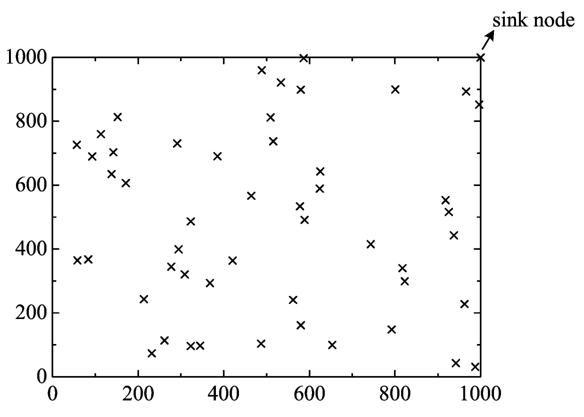


**Fig. 7　Topology of the network**

We use a heavy-load traffic scenario in the simulations, as our purpose is to evaluate the performance of S-MAC, RMAC and HMAC under such network circumstances. In the simulations, all sensor nodes send a data packet to the sink node at the same time. Moreover, the simultaneous delivery event does not happen only once, but occurs at a certain interval throughout the entire simulated time. It reveals an increase in the traffic load when the packet interval decreases.

It should be noted that we do not use a random or realistic but an extreme scenario. On the one hand, it is because HMAC is designed mainly for heavy traffic load. We have simulated HMAC in the former conditions previously, but it is quite difficult to reach extremely heavy-load traffic by using these scenarios. One the other hand, although HMAC also applies to light-load traffic, the performance differences of these MAC protocols are not very obvious. Fig. 8 illustrates the average sensor power of different MAC protocols in a realistic scenario, in which sensor nodes send packets one by one at a certain interval. Apparently, the sensor power differences of the protocols are not quite great.

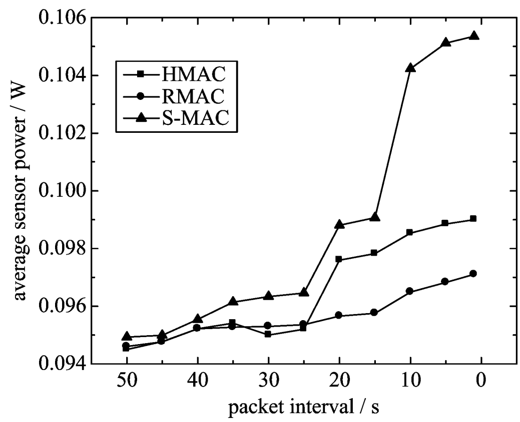


**Fig. 8　Average power sensor in the realistic scenario**

### 3.3　Results and analysis

#### 3.3.1　Energy efficiency evaluation

The energy efficiency performance of HMAC is verified in this section. Keeping the other parameters fixed, we vary the packet interval from 500 s to 1 s, observing and recording the corresponding results.

Fig. 9 illustrates the average power of sensors in different MAC protocols. When the traffic load is relatively light, for example, when the packet interval is 500 s, the energy consumptions of S-MAC, RMAC and HMAC are all small. As the interval decreases gradually, the tendencies of the three curves are all increasing monotonously.

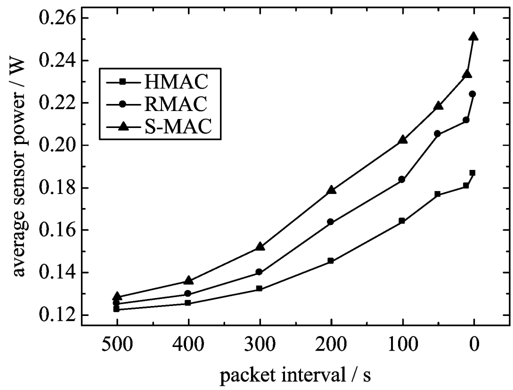


**Fig. 9 Average power of sensors**

However, HMAC has a much smaller rate of increase when compared to S-MAC and RMAC. This is because HMAC has a better mechanism to handle loss events, which is especially superior when the traffic load is heavy.

Loss events have a significant impact on S-MAC. S-MAC can forward a data packet over only one hop per cycle. Sensor nodes have to go to sleep periodically even if the traffic is busy. RMAC also pays a price for loss events. It is a waste of energy that multiple nodes along downstream wait for a packet, which is already dropped and will not be retransmitted in the current cycle until timeout. Before the retransmission starts, all nodes in the network have to be synchronized, which also causes extra energy consumption. On the contrary, it costs HMAC much less energy if a packet is lost, since HMAC just needs to keep the two communicating sensors awake for a short period until the retransmission.

With the increase in the traffic load, the frequency of packet loss rises, and the proportion of the energy consumption caused by loss events to total energy consumption enlarges. In this case, the advantage of HMAC becomes more and more obvious. As plotted in Fig. 9, HMAC shows a reduction of energy consumption of 16.6% over RMAC and 25.7% over SMAC when the packet interval is 1 s.

3.3.2 Latency evaluation

In this section, we focus on the observation of the end-to-end delivery latency of HMAC when it suffers heavy-load traffic. For this purpose, here we also vary the packet interval from 500 s to 1 s.

We offer the latency results in Fig. 10. It is observed that both RMAC and HMAC outperform S-MAC, since S-MAC does not support multihop packet delivery within a single cycle but RMAC and HMAC do. A data packet that needs to be delivered over various hops in SMAC suffers significant latencies generated at the relay nodes.

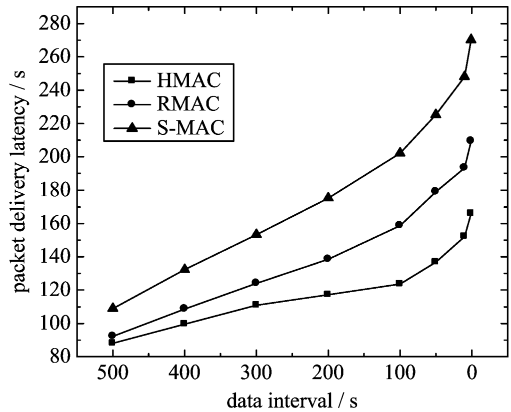


**Fig. 10 End-to-end delivery latency**

Furthermore, our results indicate that HMAC also shows an improvement over RMAC, due to better handling of loss events. Retransmissions in RMAC always occur in the next cycle, which leads to a considerable latency composed of two parts. One is an unfixed latency generated in the current cycle, since the nodes along downstream have to unnecessarily wait for the lost packets until timeout. The earlier the loss event happens, the greater the latency will be. The other one is a fixed latency generated in the Sync period of the next cycle, as all sensor nodes have to be synchronized before the retransmission starts. In contrast, loss events have much less impact on HMAC, due to a much shorter waiting period from the packet loss to the retransmission. Compared with S-MAC and RMAC, HMAC generally reduces the delivery latency by 12.6% and 28.9%, respectively.

## 4 Related work

We discuss some previous approaches designed for wireless sensor networks here.

S-MAC[1] is one of the first duty-cycle MAC protocols. S-MAC saves more energy than previous works, due to the periodic listen/sleep mechanism. T-MAC[3] dynamically ends the active period if there is no traffic in the network. Though duty cycling is energy efficient, significant latency is caused during multihop packet delivery, since a data packet can be delivered over only one hop in a cycle.

S-MAC with adaptive listening[2] overcomes the limitation. When a node overhears an RTS or a CTS, it wakes up for a short period when the communication ends. If this node is the next hop node, it can receive the packet from its neighbor node immediately. As a result, a data packet can be delivered up to 2 hops in a single cycle.

RMAC[4] improves the end-to-end latency by using the PION multihop forwarding mechanism. Similar to RMAC, DW-MAC[5] employs SCH frames for scheduling nodes to wake up during the Sleep period of a cycle. With a one-to-one proportional mapping function, DW-MAC allows nodes to wake up on demand, and ensures that data transmissions do not collide at their intended receivers. If a loss event happens, the retransmission has to wait until the next cycle for both RMAC and DW-MAC.

In contrast to the above techniques, HMAC is unique, as it is suitable for heavy-load traffic scenarios. HMAC can forward packets over multiple hops in a single cycle. Moreover, it reduces the energy consumption caused by loss events. These features are achieved by using its scheduling frames, which handles collisions well.

## 5 Conclusion

In this paper, we presented HMAC, a new energy efficient MAC protocol designed to reduce the unnecessary energy consumption caused by loss events, particularly in a heavy-load traffic bursty scenario. HMAC employs a unique way to handle loss events through the use of $R_f/R_p$ frames. These scheduling frames contain some cross-layer routing information. HMAC exploits the cross-layer information to achieve efficient multihop delivery in a single cycle, and guarantee that loss events do not impact unrelated downstream nodes. Therefore, extra energy consumption introduced by loss events can be mitigated. The simulation results show that HMAC outperforms S-MAC and RMAC in heavy traffic load situations, with higher power efficiency and lower delivery latency.

## Appendix

**Proof of** $(E_h^r)_R > (E_h^r)_H$

Since $W_1 \gg W_5, W_2 \gg W_5, W_3 \gg W_5$,

$$(E'_h)_R - (E'_h)_H = \pi_1(W_1 - W_5) + \pi_0\pi_1(W_2 - W_5) + \pi_0^2 \cdot \pi_1(W_3 - W_5) - \pi_0^3\pi_1(4\varepsilon + W_5) > 0 \quad (20)$$

Thus the difference between the first term of Eq. (11) and that of Eq. (15) is

$$D_1 = (1-\pi_0^3)^x \cdot x \cdot (E'_h)_R - (1-\pi_0^4)^x \cdot x \cdot (E'_h)_H > (1-\pi_0^4)^x \cdot x \cdot (E'_h)_R - (1-\pi_0^4)^x \cdot x \cdot (E'_h)_H = (1-\pi_0^4)^x \cdot x \cdot [(E'_h)_R - (E'_h)_H] > 0 \quad (21)$$

Then we consider the second term of Eq. (11) and that of Eq. (15), respectively:

$$\sum_{i=1}^{x}(1-\pi_0^3)^{i-1}\pi_0^3[(i-1)(E'_h)_R + 3\varepsilon + W_4] \approx \pi_0^3(3\varepsilon + W_4) + [\pi_0^3 - O(\pi_0^6)] \cdot [(E'_h)_R + 3\varepsilon + W_4] + [\pi_0^3 - O(2\pi_0^6) + O(\pi_0^9)] \cdot [2(E'_h)_R + 3\varepsilon + W_4] + \cdots = \sum_{i=1}^{x}\pi_0^3(i-1)[(E'_h)_R + 3\varepsilon + W_4] = \pi_0^3 \cdot (3\varepsilon + W_4) \cdot x + \frac{x(x-1)\pi_0^3(E'_h)_R}{2} \quad (22)$$

$$\sum_{i=1}^{x}(1-\pi_0^4)^{i-1}\pi_0^4[(i-1)(E'_h)_H + 4\varepsilon] \approx \pi_0^4 \cdot 4\varepsilon + [\pi_0^4 - O(\pi_0^8)] \cdot [(E'_h)_H + 4\varepsilon] + [\pi_0^4 - O(2\pi_0^8) + O(\pi_0^{12})] \cdot [2(E'_h)_H + 4\varepsilon] + [\pi_0^4 - O(3\pi_0^5) + O(3\pi_0^9) + O(\pi_0^{12})] \cdot [3(E'_h)_H + 4\varepsilon] + \cdots = \sum_{i=1}^{x}\pi_0^4[(i-1)(E'_h)_H + 4\varepsilon] =$$

$$\pi_0^4 \cdot 4\varepsilon \cdot x + \frac{x(x-1)\pi_0^4 (E'_{\mathrm{h}})_{\mathrm{H}}}{2} \tag{23}$$

Therefore, the difference between the second term of Eq. (11) and that of Eq. (15), namely, the difference between Eq. (22) and Eq. (23), can be computed as

$$D_2 = \sum_{i=1}^{x} (1-\pi_0^3)^{i-1} \pi_0^3 [(i-1)(E'_{\mathrm{h}})_{\mathrm{R}} + 3\varepsilon + W_4] - \sum_{i=1}^{x} (1-\pi_0^4)^{i-1} \pi_0^4 [(i-1)(E'_{\mathrm{h}})_{\mathrm{H}} + 4\varepsilon] = (\pi_0^3 \cdot 3\varepsilon \cdot x - \pi_0^4 \cdot 4\varepsilon \cdot x) + \frac{x(x-1)}{2} \cdot [\pi_0^3 (E'_{\mathrm{h}})_{\mathrm{R}} - \pi_0^4 (E'_{\mathrm{h}})_{\mathrm{H}}] + \pi_0^3 \cdot W_4 \cdot x > 0 \tag{24}$$

According to Eq. (21) and Eq. (24), we can obtain the following:

$$(E_{\mathrm{h}}^{\mathrm{r}})_{\mathrm{R}} - (E_{\mathrm{h}}^{\mathrm{r}})_{\mathrm{H}} = D_1 + D_2 > 0 \tag{25}$$

## References


[1] Ye W, Heidemann J, Estrin D. An energy-efficient MAC protocol for wireless sensor networks [C]// Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies(IEEE INFOCOM 2002). New York: IEEE, 2002:1 567-1 576.

[2] Ye W, Heidemann J, Estrin D. Medium access control with coordinated adaptive sleeping for wireless sensor networks [J]. IEEE/ACM Transactions on Networking, 2004,12(3):493-506.

[3] van Dam T, Langendoen K. An adaptive energy-efficient MAC protocol for wireless sensor networks[C]// Proceedings of the 1st International Conference on Embedded Networked Sensor Systems. Los Angeles, California, USA:ACM, 2003: 171-180.

[4] Du S, Saha A K, Johnson D B. RMAC: A routing-enhanced duty-cycle MAC protocol for wireless sensor networks[C]//Proceedins of the 26th IEEE Conference on Computer Communications (INFOCOM 2007). Anchorage, AK: IEEE, 2007:1 478-1 486.

[5] Sun Y, Du S, Gurewitz O, et al. DW-MAC: A low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks [C]// Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing. New York:ACM,2008:53-62.

[6] Bolot J C. End-to-end packet delay and loss behavior in the Internet [J]. Computer Communication Review, 1993,23(4):289-298.

[7] Wen Hao, Lin Chuang, Ren Fengyuan, et al. Retransmission or redundancy: Transmission reliability in wireless sensor networks[C]// Proceedings of 2007 IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2007). Pisa, Italy: IEEE, 2007:222-228.