Tang HW, Cao JN, Liu XF *et al.* SR-MAC: A low latency MAC protocol for multi-packet transmissions in wireless sensor networks. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 28(2): 329–342 Mar. 2013. DOI 10.1007/s11390-013-1334-4

SR-MAC: A Low Latency MAC Protocol for Multi-Packet Transmissions in Wireless Sensor Networks

Hong-Wei Tang[1] (唐宏伟), Jian-Nong Cao[2] (曹建农), *Senior Member*, *CCF*, *IEEE*, *Member*, *ACM*
Xue-Feng Liu[2] (刘雪峰), and Cai-Xia Sun[1] (孙彩霞)

[1] *School of Computer, National University of Defense Technology, Changsha 410073, China*
[2] *Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong, China*

E-mail: hwtang@nudt.edu.cn; {csjcao, csxfliu}@comp.polyu.edu.hk; cxsun@nudt.edu.cn

Received February 20, 2012; revised August 30, 2012.

Abstract Event detection is one of the major applications of wireless sensor networks (WSNs). Most of existing medium access control (MAC) protocols are mainly optimized for the situation under which an event only generates one packet on a single sensor node. When an event generates multiple packets on a single node, the performance of these MAC protocols degrades rapidly. In this paper, we present a new synchronous duty-cycle MAC protocol called SR-MAC for the event detection applications in which multiple packets are generated on a single node. SR-MAC introduces a new scheduling mechanism that reserves few time slots during the SLEEP period for the nodes to transmit multiple packets. By this approach, SR-MAC can schedule multiple packets generated by an event on a single node to be forwarded over multiple hops in one operational cycle without collision. We use event delivery latency (EDL) and event delivery ratio (EDR) to measure the event detection capability of the SR-MAC protocol. Through detailed ns-2 simulation, the results show that SR-MAC can achieve lower EDL, higher EDR and higher network throughput with guaranteed energy efficiency compared with R-MAC, DW-MAC and PR-MAC.

Keywords medium access control protocol, synchronous, duty cycle, event delivery latency, event delivery ratio

1 Introduction

With the development of wireless communication, embedded computation and sensor technology, wireless sensor networks (WSNs) are used widely in many application areas including military, industry, agriculture and environmental monitoring, and have been an active research area in the past few years. In WSNs, medium access control (MAC) protocols determine how the wireless devices use the wireless channel and assign the limited resource of wireless communication, and are fundamental protocols and key techniques in WSNs.

In the wireless MAC protocols used by traditional wireless ad hoc networks, such as IEEE 802.11[1], each wireless device needs to listen to the wireless channel at all time in order not to miss possible incoming packets, even though when there is no communication on the channel. We call this channel listening with no communication as idle listening. Idle listening is the major source of energy waste in wireless devices[2]. In WSNs, sensor nodes commonly are powered by the battery with finite capacity. Because of high density deployment and hostile deployment environment, it is infeasible to supply power by exchanging the battery. The battery becomes the key resource in WSNs. Therefore, traditional MAC protocols are not suitable for WSNs.

To reduce the energy consumption of idle listening, some MAC protocols based on the use of duty-cycle[3-4] have been presented for WSNs. In duty-cycle mechanism, the state of nodes periodically switches between active and sleeping. Nodes can transmit or receive packets in active state and shut down the radio completely in sleeping state to save energy. The energy consumption in sleeping state is ultra-low. Therefore, compared with listening to the channel at all time in traditional MAC protocols, MAC protocols based on duty-cycle for WSNs can save energy significantly.

Duty-cycle mechanism can significantly reduce energy consumption of idle listening. However, when a node wants to transmit data packets, its next hop node may be in sleeping state, which results in that the packets cannot be forwarded immediately. This situation will prolong packet delivery latency, called as sleep latency. This sleep latency becomes more and more seri-

---

Regular Paper
This work was supported the National Natural Science Foundation of China under Grant No. 61103011.

ous with the path length increasing. For some real-time applications, it cannot be tolerant. R-MAC[5], DW-MAC[6] and PR-MAC[7] all adopt cross-layer optimization to schedule packets to be forwarded over multiple hops in one operation cycle, so that packet delivery latency can be reduced. X-MAC[8] uses short strobed preamble to reduce packet delivery latency, and RI-MAC[9] and PW-MAC[10] adopt the receiver-initiated mechanism to reduce this latency.

In WSNs, one of the major applications is event detection. In many cases, such as videos, photos, and audios, a node needs long message to describe the detected event. Due to the unreliability of the wireless channel, the long message has to be re-transmitted even if only a few bits have been corrupted in the first transmission. The re-transmission of the long message will waste significant energy. Consequently, it is necessary to fragment the long message into many independent short packets when the node detects an event, and a single node may generate multiple packets to describe it after detecting an event. When an event generates multiple packets on a single node, the node maybe has to forward them during the multiple operational cycles so that the performances of existing MAC protocols degrade rapidly.

In this paper, we present a new slot-reserved, synchronous duty-cycle MAC protocol called SR-MAC aiming at event detection in WSNs. In order to reduce packet delivery latency on the multi-hop path, SR-MAC uses slot-reserved mechanism and cross-layer optimization to schedule nodes to forward the packets over the multi-hop path in the SLEEP period without collision. Furthermore, by adopting multi-packet transmission mechanism, SR-MAC schedules nodes to forward multiple packets that are generated by an event on a single node in one operational cycle. This can enhance the protocol's performance significantly.

The contributions of this paper are as follows:

- The proposed SR-MAC introduces a new slot-reserved mechanism to schedule the nodes to forward packets in the SLEEP period without collision.
- To reduce event delivery latency and improve event delivery ratio, SR-MAC schedules nodes to forward multiple packets that are generated by an event on a single node in one operational cycle.
- We evaluate SR-MAC's performance through detailed ns-2 simulation① and compare the results with other synchronous duty-cycle MAC protocols.

The rest of this paper is organized as follows. Section 2 describes the related work on synchronous duty-cycle MAC protocols for WSNs. Section 3 presents the design of SR-MAC, and Section 4 shows the performance of SR-MAC through detailed ns-2 simulation and compares it with R-MAC, PR-MAC and DW-MAC. Finally, in Section 5, we draw conclusions.

2 Related Work

Contention-based duty-cycle MAC protocols can be classified into two categories: synchronous and asynchronous duty-cycle MAC protocols. Synchronous duty-cycle MAC protocols, such as S-MAC, S-MAC-AL[11], T-MAC[12], R-MAC, DW-MAC and PR-MAC, need neighbor nodes to synchronize so that they can converge the wakeup and the sleep time. All nodes in synchronous duty-cycle MAC protocols wake up simultaneously to communicate. In contrast, asynchronous duty-cycle MAC protocols, such as B-MAC, WiseMAC[13], X-MAC, RI-MAC and PW-MAC, do not require any synchronization between neighbor nodes. Each node in these protocols decides its wakeup and sleeping time according to its own schedule. SR-MAC presented in this paper is a synchronous duty-cycle MAC protocol, so we only discuss synchronous MAC protocols in this section.

S-MAC is one of the original synchronous duty-cycle MAC protocols for WSNs. S-MAC divides the time into operational cycles and each cycle is divided into three periods: SYNC, DATA and SLEEP. Nodes in S-MAC wake up at the start of the SYNC period and broadcast special packets periodically to synchronize the clocks with neighbor nodes. In the DATA period, the nodes participating in communication exchange Request-to-Send (RTS) and Clear-to-Send (CTS) frames and switch into the low power SLEEP period after packet transmission. The nodes having no pending packets directly sleep after the DATA period until the next SYNC period. S-MAC only forwards packets over one hop in one operational cycle and packet delivery latency on multi-hop path increases significantly.

To reduce the packet delivery latency over multi-hop delivery path, S-MAC is enhanced by introducing adaptive listening, called as S-MAC-AL. With adaptive listening, if a node overhears the communication from other nodes (e.g., RTS or CTS) during the DATA period, it wakes up for a short time when the overheard communication finishes. If this node is exactly the next hop node along the multi-hop path of packet delivery, its upstream node can forward the packet to it rather than wait for the DATA period in the next operational cycle to initiate the forwarding. Fig.1 shows the example of the operation of S-MAC-AL. Node A sends a packet to node B that has a next hop node of C. During the DATA period, C overhears CTS that is from B to A. C goes to sleep immediately at the beginning of

---

① http://www.isi.edu.nsnam/ns/, Feb. 2012.

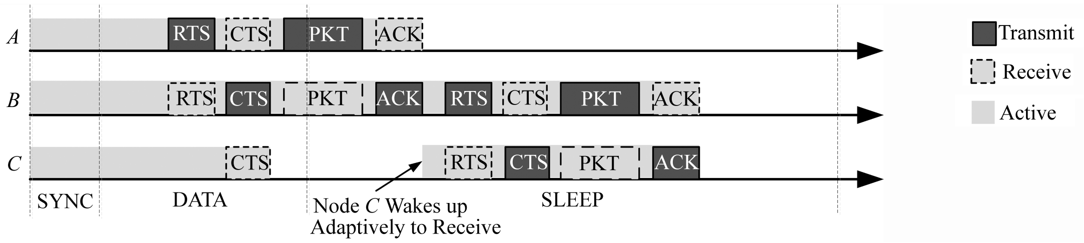


Fig.1. Overview of S-MAC and S-MAC-AL.

the SLEEP period. According to the information in CTS overheard by C, it wakes up again to listen to the channel after the acknowledgement (ACK) from B to A. B can immediately forward the data packet (PKT) to C at this cycle rather than waiting until the next operational cycle. S-MAC-AL can deliver a packet up to two hops per operational cycle at most. Though S-MAC-AL can reduce packet delivery latency, all neighbor nodes of B overhear CTS from B, so these nodes have to wake up again when the communication between A and B finishes, even though they are not the next hop node of the packet.

The duration of the DATA period is fixed in S-MAC and S-MAC-AL protocols. Even though the nodes have not any communication in current operational cycle, they also have to listen to the channel until the start of the SLEEP period. Because the nodes in WSNs have no communication in most of time, idle listening of the DATA period wastes abundant energy. This fixed DATA period is not suitable for light traffic load. T-MAC is primarily designed to shorten the DATA period when no traffic is around the nodes, so that the nodes without communication can preserve more energy. Its principle is that nodes go to sleep if they cannot detect any specified events in TA which is the minimum idle listening time of nodes in a cycle. Although T-MAC can preserve more energy than S-MAC under light traffic load, it also only delivers a packet up to two hops within one operational cycle and cannot further reduce multi-hop delivery latency of the packet.

Some other approaches are proposed to reduce deliver latency. However, they make some specific assumptions on the communication pattern. For example, D-MAC[14] reduces data delivery latency only for data gathering tree. Lu *et al.*[15] discussed how to minimize end-to-end delivery latency for a tree or a ring network. Keshavarzian *et al.*[16] analyzed latency for different wakeup patterns and proposed the multi-parent technique to improve the protocol's performance under the assumption that nodes can have more than one single parent.

R-MAC presents a different approach to reduce packet delivery latency in multi-hop forwarding. R-MAC uses cross-layer routing information to schedule the nodes on the packet's forwarding path to wake up one by one to reduce packet delivery latency over multi-hop delivery path. R-MAC can forward the packet over more hops than S-MAC-AL and the nodes only on the packet's forward path will wake up in the SLEEP period, so that R-MAC can achieve lower packet delivery latency and energy consumption compared with S-MAC-AL. Fig.2 gives an overview of R-MAC. R-MAC replaces RTS/CTS by the special pioneer frame, called as PION. In an operational cycle, PION is forwarded over multiple hops during the DATA period to inform nodes B and C when to wake up to receive or transmit the data packet during the SLEEP period. The process of PION forwarding goes on till the DATA period is over or PION arrives at the final destination node, so maximum hop counts over which R-MAC can forward a packet in one operational cycle is limited by the duration of the DATA period. According to hop counts carried in PION, nodes that are on the data forwarding path calculate their wakeup time during the SLEEP period using (1):

$$T_{\text{wakeup}}(i) = (i-1) \times (durDATA + SIFS + durACK + SIFS), \qquad (1)$$

where $durDATA$ and $durACK$ are the duration of the

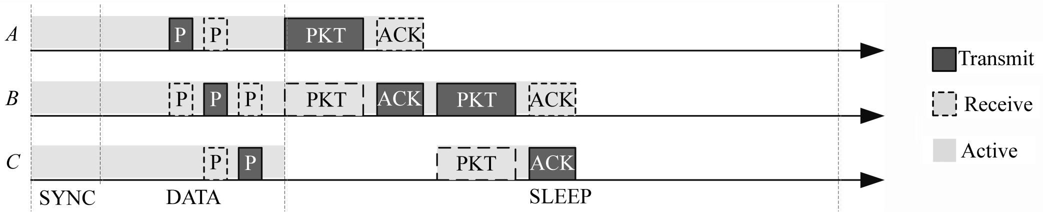


Fig.2. Multiple hops forwarding of R-MAC. P: PION.

data packet and the ACK frame transmissions respectively, and *SIFS* is short inter-frame spacing.

Because the packet transmissions always begin at the start of the SLEEP period, hidden terminal nodes that have succeeded in contending the channel and sending PION during the DATA period will cause data packet collisions. The nodes have to re-transmit the collided packets. These re-transmissions for collisions will result in significant energy waste and prolong the packet delivery latency.

DW-MAC resolves the problem that hidden terminal nodes collide when transmitting data packets in R-MAC by using one-to-one mapping to schedule nodes to wake up in the SLEEP period. Fig.3 describes the scheduling approach in DW-MAC. According to (2), based on when the nodes succeed in contending the channel during the DATA period and the duration of the SCH transmission, DW-MAC determines when the nodes should wake up to communicate with their neighbor nodes during the SLEEP period and how long they can occupy the channel. In this example, node S wants to transmit a data packet to node R. S first contends for channel access and transmits an SCH during the DATA period. Suppose transmission of the SCH starts T_1 time units after the beginning of the DATA period. Based on T_1 and the duration of the SCH transmission, T_3, S and R will both schedule their wakeup time to T_2 from the beginning of the following SLEEP period, and will agree on a maximum wakeup time duration of T_4, based on the ratio between $T_{\rm DATA}$ and $T_{\rm SLEEP}$, as shown in Fig.3. By one-to-one mapping function, packet transmissions during the SLEEP period in DW-MAC will not collide. In addition, DW-MAC uses cross-layer routing information to schedule the packet to forward over multi-hop path, so that it can reduce packet delivery latency.

$$\frac{T_2}{T_1}=\frac{T_4}{T_3}=\frac{T_{\rm SLEEP}}{T_{\rm DATA}}. \tag{2}$$

DW-MAC resolves the problem that hidden terminal nodes collide in R-MAC. However, in one operational cycle, DW-MAC only schedules one packet to deliver during the SLEEP period. If multiple packets are generated by an event on a single node, DW-MAC has to schedule nodes to deliver multiple packets to the sink node in a few operational cycles. This increases packets delivery latency. Fig.4 describes the procedure of multi-packet transmissions in DW-MAC. Node S generates two packets for node R to describe the event when it detects an event. S and R go to sleep after transmitting the first packet in the first operational cycle, but not keep transmitting packets in the queue. In the next operational cycle, S transmits the second packet to R. Therefore, S at least needs two cycles to send all packets of the event. The more packets are generated by the event on S, the more cycles are needed by S to send all packets. The multi-packet delivery latency is significantly increased. In other words, R takes more time to receive the event.

PR-MAC derives from R-MAC and enhances R-MAC. In PR-MAC, multiple packets can be scheduled to be forwarded in one operational cycle. But PR-MAC also faces the same problem as R-MAC that hidden terminal source nodes collide when they start transmitting the packets at the beginning of the SLEEP period.

3 SR-MAC Design

In this section, we describe the design of SR-MAC protocol in detail. Subsection 3.1 summarizes SR-MAC

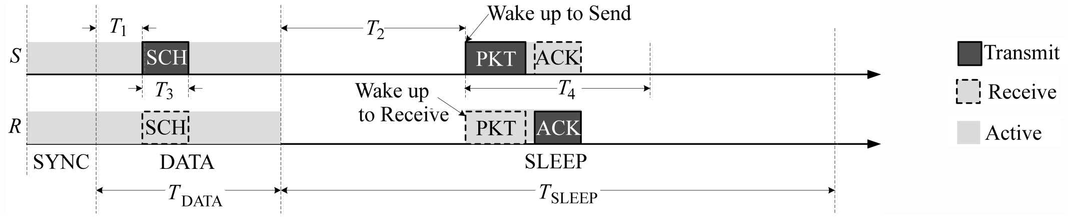


Fig.3. Schedule of DW-MAC.

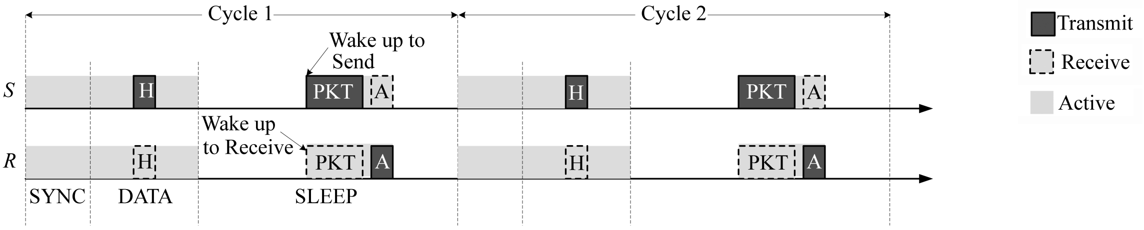


Fig.4. Multiple packets transmission in DW-MAC. H: SCH, A: ACK.

protocol. Nodes scheduling based on slot-reserved is presented in Subsection 3.2. Subsection 3.3 introduces special slot-reserved frames and Subsection 3.4 describes multiple packets transmission in SR-MAC. Finally, in Subsection 3.5, we show an optimization for packet multi-hop forwarding.

3.1 Overview

SR-MAC is a synchronous duty-cycle MAC protocol. In SR-MAC, each operation cycle is also divided into three periods: SYNC, DATA and SLEEP period. The duration of each period is denoted by T_{SYNC}, T_{DATA} and T_{SLEEP} respectively. Similar to existing synchronous duty-cycle MAC protocols, SR-MAC must use some synchronization mechanisms[17-18] to synchronize the clock of neighbor nodes in the SYNC period.

Different from existing MAC protocols, SR-MAC adopts a slot-reserved mechanism to schedule nodes to wake up to communicate with their neighbor nodes during the SLEEP period of an operation cycle. In SR-MAC, the DATA and SLEEP periods are both divided into slots. In the DATA period, a node that wants to transmit data packets contends for the channel access using a CSMA/CA protocol described in IEEE 802.11 specification. However, SR-MAC replaces RTS/CTS with a special control frame, called as the slot-reserved frame (SRF, described in Subsection 3.3). In the SLEEP period, according to the slot in which the node transmits the SRF, the neighbor nodes wake up to communicate with each other in the corresponding slot.

The advantages of SR-MAC protocol are as follows:

- SR-MAC replaces RTS/CTS with SRF. Based on the slot in the DATA period occupied by SRF transmission, the corresponding slot in the SLEEP period is reserved for packets transmission. Only nodes that participate in communication will wake up in the particular slots in the SLEEP period, which can reduce energy waste because of non-communication nodes' unnecessary wakeup.
- Slot-reserved mechanism guarantees that packet transmissions do not collide during the SLEEP period.
- When an event generates multiple packets on a single node, SR-MAC can reserve multiple slots of the SLEEP period in one operational cycle for packets transmission to reduce multi-packet delivery latency.

3.2 Slot-Reserved Mechanism

In this subsection, we discuss how to reserve the slot when one packet is only generated by an event on a single node. Subsection 3.4 will detail the situation of multiple packets generated by an event.

In SR-MAC, we divide the DATA period into M slots, called as data slots. In order to ensure that there is only one node can transmit its SRF within the nodes in its interference range, we must be careful to choose the duration of the data slot. The duration of each data slot is shown in (3), where *durCtrl* presents the duration of an SRF transmission. Especially, the value of M must satisfy that ($M \times \mathit{durDataSlot} \leqslant T_{\mathrm{DATA}}$).

$$\mathit{durDataSlot} = \mathit{durCtrl}. \tag{3}$$

In SR-MAC, the SLEEP period is also divided into M slots, called as sleep slots, and the duration of each sleep slot is shown as (4):

$$\mathit{durSleepSlot} = T_{\mathrm{SLEEP}}/M. \tag{4}$$

Algorithm 1. Slot-Reserved Algorithm in SR-MAC

The DATA and SLEEP periods are both divided into M slots. The nodes with pending packets contend the channel access during the DATA period. Assumed that the time of succeeding in contending the channel for the sender S is *TIME_NOW*, then S begins to transmit its SRF to the receiver R:

1) The sender S calculates the number of data slots based on the time of transmitting SRF (*TIME_NOW*) and the start time of the current DATA period (*CurDataTime*): $k = [(\mathit{TIME\_NOW} - \mathit{CurDataTime})/\mathit{durDataSlot}]$.
2) Once receiving the SRF from S, the receiver R reserves its k-th sleep slot to receive the packet according to the information in the SRF and sends another SRF to acknowledge S.
3) Once receiving the acknowledgement SRF from R, S also reserves its k-th sleep slot to transmit the packet.

The relationship between the data slot and sleep slot is a one-to-one mapping. Namely, if the sender transmits its SRF in the k-th data slot, it will transmit the data packet to the receiver at the k-th sleep slot. Algorithm 1 describes the slot-reserved mechanism in SR-MAC.

Fig.5 shows an example of the node's schedule based on slot-reserved mechanism in SR-MAC. For example, node S has data to node R. S contends the channel access and begins to transmit its SRF during the k-th data slot. Therefore, according to the slot-reserved mechanism described in Algorithm 1, S and R both reserve k-th sleep slot to transmit the data packet.

By adopting slot-reserved mechanism of SR-MAC, we can draw a conclusion:

Theorem 1. *The communications of any reserved sleep slots are collision-free. In other words, data transmissions by nodes that wake up during the SLEEP period do not collide at their intended receivers.*

Proof. We prove our conclusion by contradiction. We assume that nodes A and B could collide in the

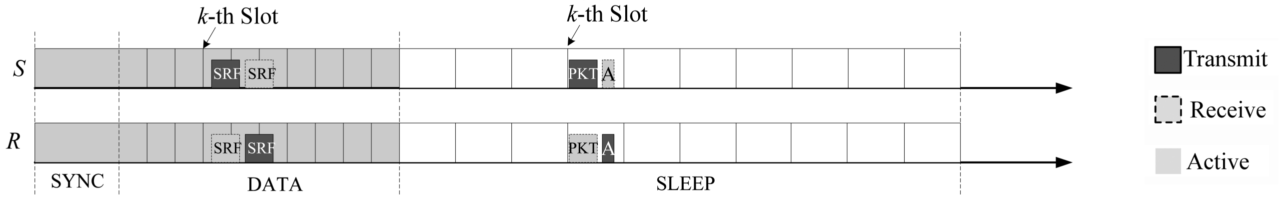


Fig.5. Slot-reserved mechanism in SR-MAC. A: ACK.

k-th reserved sleep slot. If both nodes A and B want to transmit a packet in the k-th sleep slot, they must contend the channel to send their own SRFs in the k-th data slot of the DATA period. Because A collides with B at the k-th reserved sleep slot, which means that they have collided when they sent their own SRFs at the k-th data slot. Therefore, their receivers cannot decode any SRF and they cannot reserve any sleep slots to receive the packets. Furthermore, the receivers do not transmit any acknowledgment SRFs to the senders, so A and B do not reserve their k-th sleep slots to transmit the packets, which contradicts our assumption. □

3.3 Slot-Reserved Frame

In SR-MAC, the RTS/CTS is replaced by the special SRF. In addition to the fields included in RTS/CTS, such as the sender address, receiver address and duration of the transmission, an SRF also includes the number of slots, the number of packets and the cross-layer routing information. The functions of SRF are as follows:

1) playing as a handshake frame between neighbor nodes just like RTS/CTS.

2) used to negotiate to reserve a sleep slot for packet transmission.

3) piggybacking cross-layer routing information to schedule next hop node to wake up during the SLEEP period, which will be described in Subsection 3.5.

An SRF serves as either a scheduling request or a scheduling acknowledgement. When the receiver node receives an SRF, it reserves a sleep slot for receiving packets according to the information included in the SRF and sends an acknowledgement SRF back to the sender. Once the sender receives the acknowledgement SRF from the receiver, the sender also reserves specified sleep slot for packet transmission. For multi-hop forwarding, in addition to a scheduling request or acknowledgement, an intermediate node sends an SRF that schedules the next downstream node to forward the packet.

3.4 Multiple Packets Transmission

The overhead of re-transmission of the long message is very high and the common size of a packet is usually set as 50 or 100 bytes in WSNs. Therefore, the long message is commonly fragmented into many independent short packets when the node detects an event and nodes need transmit multiple packets to describe this event.

As described in Subsection 3.2, SR-MAC divides both the DATA period and SLEEP period into M slots. According to the radio parameter and the size of an SRF, the duration of the SRF transmission can be calculated as 14.2 ms (details are shown in Subsection 4.1), that is $durCtrl = 14.2\,\text{ms}$. As the configuration of DW-MAC's simulation, if we set $T_{\text{DATA}} = 142.0\,\text{ms}$ and $T_{\text{SLEEP}} = 3\,945.0\,\text{ms}$, the DATA period of SR-MAC can be divided into 10 data slots. Consequently, the duration of each sleep slot can be calculated as 394.5 ms that is enough to send more than 400 bytes based on the packet transmission latency calculated by R-MAC. If the SLEEP period is still divided into M sleep slots, only one short packet of an event can be transmitted in one sleep slot and most time of each sleep slot is wasted, so the multi-packet delivery latency increases significantly.

To schedule the node to transmit multiple short packets generated by an event on it, SR-MAC enhances the approach of dividing the SLEEP period. Firstly, the SLEEP period is divided into N frames, and then each frame is divided into M slots, so that there are $N \times M$ sleep slots in the SLEEP period. However, in order to optimize for multi-hop forwarding described in Subsection 3.5, each node can only transmit one data packet in each frame. Therefore, the maximum number of data packets transmitted by a node is N at most in one operational cycle, but not $N \times M$. For any node u, its assigned frame-slot pair (F_u, S_u) determines in which sleep slot node u can send or receive packets. For example, if a node's assigned frame-slot pair is (n, m), this node will wake up to the communicate with its neighbor nodes at the m-th slot of the n-th frame. In a sleep slot, only one data packet can be transmitted. Each sleep slot's duration is shown in (5), where *durData* presents the duration of a data packet transmission, *SIFS* presents the short inter-frame spacing between two continuous packets transmissions, and *durACK* presents the duration of sending an ACK frame.

$$durSleepSlot = durData + SIFS + durACK. \tag{5}$$

The number of frames in the SLEEP period, N, is determined by the duration of one sleep slot and the number of sleep slots M in each frame. N can be calculated by (6) and determines the maximum number of short packets transmitted by a node in one operational cycle.

$$N = \frac{T_{\mathrm{SLEEP}}}{M \times \mathit{durSleepSlot}}. \tag{6}$$

Algorithm 2 describes how SR-MAC reserves frame-slot pair when a node has multiple packets to be transmitted. If the sender S wants to send packets, it contends the channel in the DATA period at first. S succeeds in contending the channel and sends an SRF to the receiver R. We assume that S sends its own SRF at the k-th data slot. S and R will reserve k-th sleep slot of each frame in the SLEEP period to communicate with each other. In other words, the k-th sleep slot of the first frame is for the first packet, the k-th sleep slot of the second frame is for the second packet, and so on. If the number of packets that need to be transmitted by S is less than N, S and R go to sleep after transmitting all packets and no more wake up in the corresponding sleep slot of the following frames to save energy. On the other hand, the number of time frames in the SLEEP period decides that S can transmit N packets to R at most in one operational cycle. Even though S has more than N data packets, it has to wait until the next cycle.

Algorithm 2. Frame-Slot Pair Assignment Algorithm for Multiple Packets Transmission in SR-MAC

The DATA period is divided into M slots. The SLEEP period is divided into N frames and each frame is divided into M slots. The nodes with pending packets contend the channel access. We assume that the time of succeeding in contending the channel is *TIME_NOW*, and then S begins to transmit SRF to the receiver R:

1) The sender S calculates the number of data slots based on the time of SRF transmission (*TIME_NOW*) and the start time of the current DATA period (*CurDataTime*): $k = [(\mathit{TIME\_NOW} - \mathit{CurDataTime})/\mathit{durDataSlot}]$.
2) Once receiving S' SRF, the receiver R assigns its frame-slot pairs to receive packets according to the information in the SRF and sends another SRF to acknowledge the SRF from S. For the first packet, $F_R = 1$, $S_R = k$, and its frame-slot pair is $(1, k)$. And for the second packet, $F_R = 2$, $S_R = k$, its frame-slot pair is $(2, k)$. The maximum number of packets received by R is N, so the maximum value of F_R is N.
3) S also reserves the corresponding frame-slot pairs to transmit the packets once receiving acknowledgement SRF from R.

Fig.6 shows an example of multiple packets transmission in SR-MAC. When detecting an event, node s generates two packets ($D1$ and $D2$) to send to node r. s sends an SRF to r during the k-th data slot. After receiving the SRF from s, r reserves the k-th sleep slots of the first and the second frames to receive these two packets from s. s will reserve the same frame-slot pairs as r to send the data packets to r. According to the described scheduling approach, s can send these two packets to r in one operational cycle. Compared to the example shown in Fig.4, SR-MAC can reduce multi-packet delivery latency. Furthermore, if the number of packets generated by the event on s is less than N, s always transmits all packets to r in one operation cycle in SR-MAC. According to the frame-slot pair reserved mechanism, s and r can use the k-th sleep slot of each frame to communicate with each other. However, in our example, there are only two packets to be transmitted. In order to save energy, it is unnecessary to wake up again in the k-th sleep slot in the successive frames of the second frame.

We also can prove that the transmission of multiple packets does not collide in SR-MAC. According to the proof in Subsection 3.2, we can draw a conclusion that the packets transmission is non-collision in the first frame. Based on the frame-slot assignment mechanism described in Algorithm 2, the following frames are like the first frame, that is to say that the nodes participating in communication in each slot of the following frames are the same as that in each slot of the first frame. Therefore, the packet transmission is non-collision in the following frames.

3.5 Optimization for Multi-Hop Forwarding

In order to reduce packet delivery latency over multi-

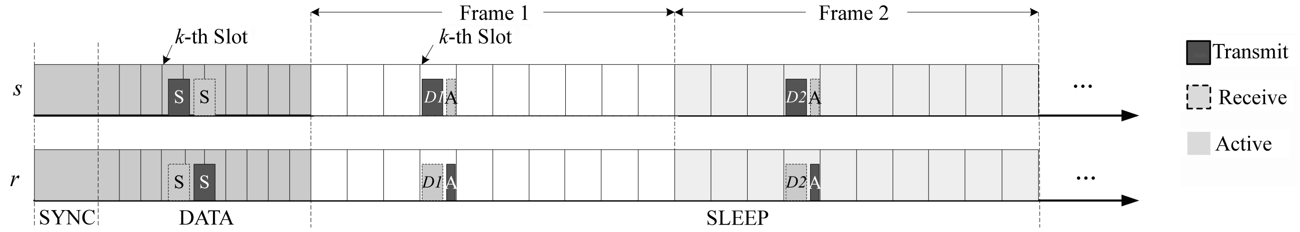


Fig.6. Multiple packets transmission in SR-MAC. S: SCR. A: ACK.

hop path, SR-MAC uses cross-layer routing information like R-MAC to schedule the packet forwarding over many hops during one operational cycle. Fig.7 illustrates the multi-hop forwarding in SR-MAC. In this example, node a has two packets to node c through node b. a firstly send an SRF to node b, and then a and b reserve the k-th sleep slot of the first and second frames to communicate. Based on cross-layer routing information, b finds that c is its next hop node. Waiting for SIFS after receiving the SRF from a, b sends its own SRF. The SRF from b not only confirms the SRF just from a but also requests c to reserve sleep slots to receive packets from b. c reserves the j-th slot of the first and second frames during the SLEEP period. Based on this mechanism, SR-MAC can forward multiple packets generated by an event on a single node over multiple hops path, so it reduces the multi-packet delivery latency over multi-hop path.

4 Simulation and Evaluation

To evaluate the performance of SR-MAC, we use Version 2.29 of the ns-2 network simulator to simulate SR-MAC and compare it with R-MAC, PR-MAC and DW-MAC.

4.1 Simulation Environment

Table 1 summarizes some key network parameters used in our simulation. Each node equips with a single omni-directional antenna, and we use the common ns-2 combined free space and two-ray ground reflection radio propagation model. For the parameter of the radio, we do not use the typical values for Mica2 radios (CC1000)[19], but use the default values in the ns-2 package. They were used also in the simulations of R-MAC and DW-MAC in the previous work. The transition time of the CC1000 radio between sleep and active states is around 2.47 ms, but the state transition power is not available in the datasheet. Because these MAC protocols used to compare have similar state transitions, we ignore the state transition power in our simulation. In evaluating sensor node power consumption, we focus on the radio's energy consumption, so ignore energy consumed by other modules of the sensor node, such as CPU, memory and so on[20].

Table 1. Network Parameters

Parameter	Value	Parameter	Value
Bandwidth	20 Kbps	Tx Range	250 m
Tx power	0.5 W	carrier sensing range	550 m
Rx power	0.5 W	contention window	64 ms
Idle power	0.45 W	Size of RTS/CTS/ACK	10 B
Sleep power	0.05 W	Size of SCH/SR	14 B
SIFS	5 ms	Size of data packet ($size_{\text{pkt}}$)	50 B
DIFS	10 ms	Channel encoding ratio	2

In our simulation, the duty cycle of all MAC protocols simulated is kept the same value of 5%. Table 2 shows the durations of T_{SYNC}, T_{DATA}, T_{SLEEP} and T_{cycle} in all protocols, where T_{cycle} means the duration of a cycle, i.e., $T_{\text{cycle}} = T_{\text{SYNC}} + T_{\text{DATA}} + T_{\text{SLEEP}}$.

Table 2. Duty Cycle Configuration

T_{SYNC} (ms)	T_{DARA} (ms)	T_{SLEEP} (ms)	T_{cycle} (ms)
55.2	142.0	3747.8	3945.0

Because of the high overhead for the long message re-transmission, a long message is generally fragmented into many independent short packets. We fix the packet size of 50 bytes in SR-MAC. The transmission latencies for different types of packets are shown in Table 3, which are calculated as (7), where *durPkt* means transmission latency of the data packet, and we use the 5

Table 3. Transmission Latency for Different Types of Packets

Type of Packets	Size of Packets (B)	Tx Latency (ms)
RTS/CTS/ACK	10	11.0
PION	14	14.2
SCH	14	14.2
SRF	14	14.2
DATA	50	43.0

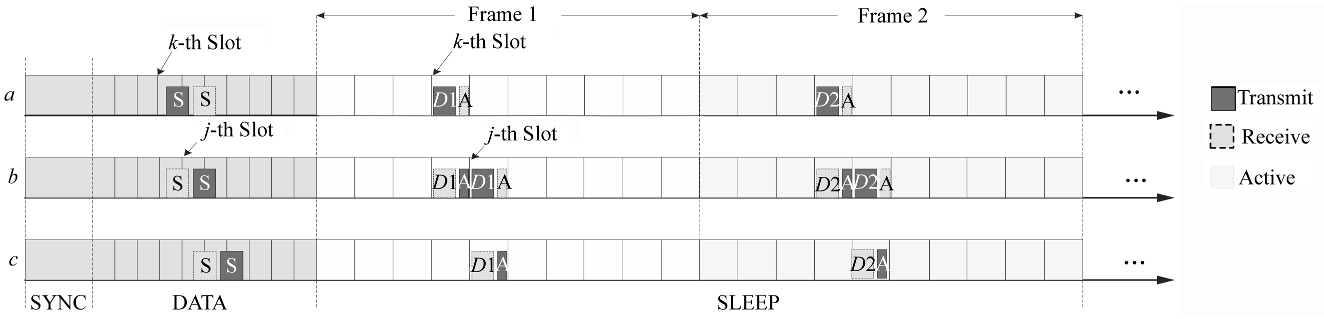


Fig.7. Multiple hops delivering in SR-MAC. S: SRF, A: ACK.

bytes for the preamble size p and the default value of $encode\_ratio$ is 2.

$$durPkt = \frac{p + size_{\rm pkt} \times encode\_ratio}{\rm Bandwidth} + 1\,{\rm ms}. \quad (7)$$

In our simulation, based on the duty cycle configuration shown in Table 2 and the transmission latencies for different types of packets shown in Table 3, the duration of the data slot is 14.2 ms and the DATA period can be divided into 10 slots. The duration of the sleep slot is 64 ms, the SLEEP period can be divided into 5 frames and each frame includes 10 sleep slots.

To evaluate the performance of SR-MAC in different network topologies, two types of scenarios are used: chain and realistic. Simple chain scenario helps us to analyze SR-MAC's performance clearly in basic multihop delivery. Simulation of realistic topology can show the performance of SR-MAC in large-scale deployment.

Fig.8 gives an example of a chain scenario. All nodes are equally spaced in a straight line and the neighbor node is placed 200 m apart to compose a chain. In our simulation, we use 21 nodes that compose a chain with 20 hops. In the chain scenario, a constant bit rate (CBR) is used to generate the traffic loads. Node 0 attached with a CBR is the source node and the CBR generates the events (packets) periodically. Node 21 is the sink node of all packets.

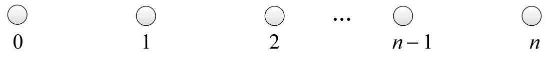


Fig.8. Chain scenario with n nodes.

In the realistic network scenario, the network consists of 100 sensor nodes, including a sink node. As shown in Fig.9, 100 sensor nodes are uniform randomly deployed in a 1 000 m by 1 000 m square area. The sink node locates at the top right corner of the square and its coordinate is (1 000, 1 000). The data packets generated by sensor nodes are all transmitted to the sink node through multiple hops delivery. In the realistic scenario, a random correlated-event (RCE)[11] that is derived from a correlated-event workload[21] is used to simulate random events. The nodes detecting the event will generate multiple packets to describe the event. In our simulation, RCE randomly selects a coordinate (x, y) in the square area and generates an event. If the sensing radius of a node is RD, the nodes within the circle centered at (x, y) with radius RD can detect the event and all of them will generate packets. With sensing radius increasing, more nodes will detect the event simultaneously and the traffic loads become heavier. Table 4 shows the average number of nodes that detect the event with different sensing radii. In our simulation, we keep the default sensing radius of 200 m and REC randomly generates an event every 200 s.

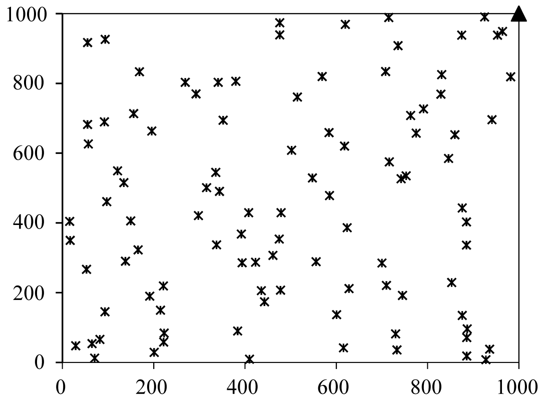


Fig.9. Realistic scenario with 100 nodes.

Table 4. Average Number of Nodes Detecting the Event Under Different Sensing Range

Range (m)	100	150	200	250	300	350
Average No. Nodes	2.9	6.2	10.6	15.8	21.6	28.1

To simulate the situation that a single node generates multiple packets when detecting an event, we keep the packet size of 50 bytes used by the user datagram protocol (UDP) of application layer. Therefore, through varying the packet size of CBR or RCE, we can control the number of packets generated by UDP protocol.

4.2 Measure Metrics

For the event detection applications, packet delivery latency (PDL) and packet delivery ratio (PDR) cannot reflect well the capability of event detection in WSNs. Therefore, in SR-MAC, we introduce event delivery latency (EDL) and event delivery ratio (EDR) to measure the capability of event detection[22]. The definitions of EDL and EDR are as follows:

- *Event Delivery Latency* (EDL). Node S detects an event and generates $N_{\rm pkt}$ packets to describe the event at the time T_0. If the sink node receives all packets of the event at the time T_1, we define EDL as $T_1 - T_0$.
- *Event Delivery Ratio* (EDR). EDR is the ratio of the number of events succeeding in being received by the sink node to the number of events detected by the source nodes. Only if the sink node receives all packets of an event, we think that the sink node succeeds in receiving this event.

To some extent, EDL and EDR can also reflect the network's PDL (packet delivery latency) and PDR (packet delivery ration). However, EDL and EDR are more suitable for event-based WSNs, because they reflect well the capability of event detection.

4.3 Event Delivery Latency Evaluation

In this subsection, we evaluate the EDL of SR-MAC. Each source node generates $N_{\rm pkt}$ packets for the sink node to describe the detected event. EDL is the duration of receiving all $N_{\rm pkt}$ packets generated by an event on a single node. In the chain scenario, the CBR attached on the source node generates an event every 50 s. In the realistic scenario, RCE generates an event every 200 s. The node detecting the event generates a message to describe it. In both of these scenarios, the message size of CBR or RCE varies from 50 bytes to 400 bytes. Because the packet size of UDP protocol keeps 50 bytes, UDP protocol fragments the messages generated by CBR and RCE into multiple packets with the size of 50 bytes, so the number of packets varies between 1 and 8.

Fig.10 shows the EDLs of all protocols with different number of data packets in the chain scenario. Because PR-MAC and SR-MAC can deliver multiple data packets in a single operational cycle, SR-MAC outperforms R-MAC and DW-MAC when $N_{\rm pkt} \geqslant 2$. Especially, when $N_{\rm pkt} = 8$, the EDL of SR-MAC is only 25.7 s and is reduced by about 94% and 50% compared with R-MAC and DW-MAC respectively. According to our simulation parameters, SR-MAC can transmit five data packets in a single cycle, so that the EDL of SR-MAC increases very slowly when $N_{\rm pkt} \leqslant 5$. SR-MAC's EDL has an obvious increase when $N_{\rm pkt} = 6$. In addition, even though in the very simple chain scenario, the EDL of SR-MAC is a little less than that of PR-MAC. The reason is that PR-MAC still suffers from the light collision at the beginning of the SLEEP period, so it has to re-transmit some collided packets.

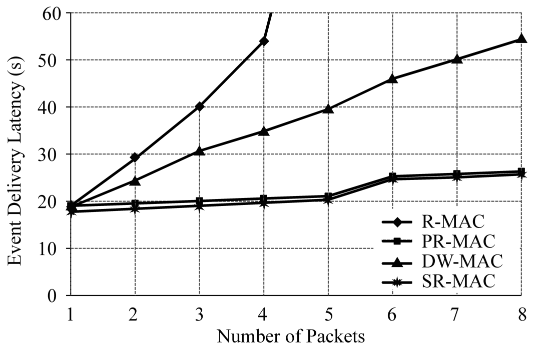


Fig.10. EDL of the chain scenario.

Fig.11 gives the EDLs of R-MAC, PR-MAC, DW-MAC and SR-MAC in the realistic scenario. With the number of packets generated by the event increasing, the EDLs of R-MAC and DW-MAC increase obviously. When $N_{\rm pkt} = 8$, the EDLs of R-MAC and DW-MAC both exceed 500 s. In the realistic scenario, there are few sensors to detect the event and transmit the data packets simultaneously, so that PR-MAC's performance degrades because of the heavier collision at the beginning of the SLEEP period. Therefore, the EDL of SR-MAC outperforms PR-MAC obviously under this situation. When $N_{\rm pkt} = 8$, the EDL of SR-MAC is 47.74 s, which is only as 42% as that of PR-MAC.

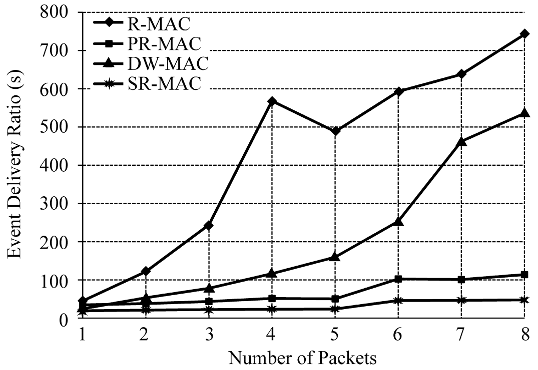


Fig.11. EDL of the realistic scenario.

4.4 Event Delivery Ratio Evaluation

We evaluate the EDR of SR-MAC in this subsection. In the chain scenario, as shown in Fig.10, DW-MAC can deliver all packets of an event within 55 s, so CBR generates an event every 20 s to evaluate EDR under heavy traffic loads. In the realistic scenario, the EDL of DW-MAC is more than 200 s when $N_{\rm pkt} = 6$. Therefore, RCE still keeps to generate an event every 200 s in the realistic scenario.

The simulation results in the chain scenario are shown in Fig.12. The EDR of R-MAC drops down from 100% to 64.2% sharply when $N_{\rm pkt} = 3$. When $N_{\rm pkt} = 8$, the EDR of R-MAC only remains 10.5%. For DW-MAC, its EDR reduces from 100% to 90.5% when $N_{\rm pkt} = 5$ and is 13.7% that is little higher than

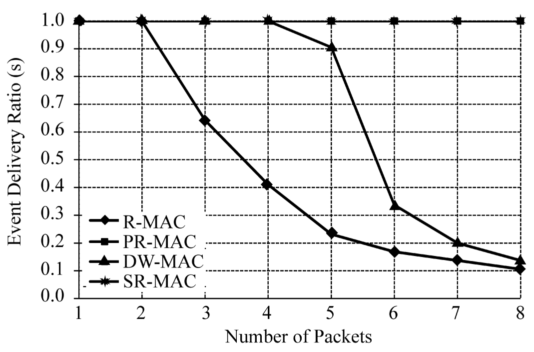


Fig.12. EDR of the chain scenario.

R-MAC when $N_{\rm pkt} = 8$. Under ultra-heavy traffic loads, the EDRs of R-MAC and DW-MAC both are very low, because they only have limited capability of packets forwarding in one single cycle. However, the EDRs of PR-MAC and SR-MAC are not effected on the number of packets generated by an event on a single node and their EDRs both maintain 100%.

Fig.13 shows the EDRs of all protocols in the realistic scenario. The EDR of R-MAC is always less than 100% and reduces sharply with the number of packets increasing. When $N_{\rm pkt} = 8$, its EDR is only 9%. Just like the chain scenario, DW-MAC's EDR drops from 100% to 90.52% when $N_{\rm pkt} = 5$. However, because the average hops of delivery path in the realistic scenario is less than in the chain scenario and RCE generates an event per 200 s, DW-MAC's EDR is still 58.4% in the realistic scenario when $N_{\rm pkt} = 8$, which is higher than in the chain scenario. Due to PR-MAC's schedule mechanism, the hidden sensors in PR-MAC may collide when starting to transmit the data packets simultaneously. The EDR of PR-MAC begins to decrease when $N_{\rm pkt} = 6$ and is about 88% when $N_{\rm pkt} = 8$. Same as the chain scenario, the EDR of SR-MAC always keeps 100% and is not effected by the number of packets.

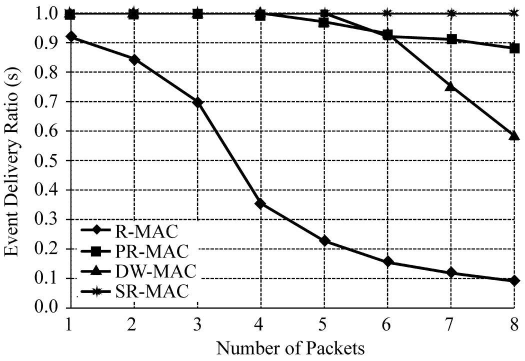


Fig.13. EDR of the realistic scenario.

4.5 Energy Consumption Evaluation

In this subsection, we evaluate the average sensor energy consumption of SR-MAC. The average sensor energy consumption is calculated by (8), where $E_{\rm sum}$ is the total energy consumption of the whole network and $N_{\rm nodes}$ presents the total number of nodes in the network. In both scenarios, the simulation runs for 2 000 seconds of simulated time.

$$E_{\rm avg} = \frac{E_{\rm sum}}{N_{\rm nodes}}. \qquad (8)$$

Fig.14 shows the average sensor energy consumption in the chain scenario. With the number of packets increasing, the average sensor energy consumption for all protocols is increased. The energy consumption of PR-MAC, DW-MAC and SR-MAC is far less than that of R-MAC and the gap is more obvious when the number of packets is increased. When $N = 8$, the energy consumption in R-MAC is as 3.82 times and 3.84 times as that of DW-MAC and SR-MAC respectively. The main reason is that the nodes always start to transmit the packets at the beginning of the SLEEP period, which causes the hidden terminal collisions and the nodes have to re-transmit the collision packets. Therefore, R-MAC wastes more energy than DW-MAC and SR-MAC because of collision. With the increase of traffic loads, the probability of the collision will be greater. Just like R-MAC, PR-MAC also suffers from the collision, so PR-MAC consumes more energy that DW-MAC and SR-MAC. Nevertheless, PR-MAC transmits multiple packets in a single cycle so that it can alleviate the effects of the collision. We find that the energy consumption of SR-MAC is a little less than that of DW-MAC. SR-MAC can use a single SRF to schedule multiple packets to be transmitted in one operational cycle, but DW-MAC sends an RTS/CTS for each packet. In our simulation, the energy consumption of idle listening is less than that of sending/receiving, so the energy consumption of SR-MAC is a little less than that of DW-MAC.

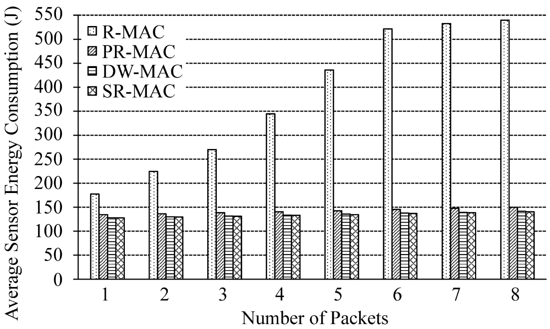


Fig.14. Average sensor energy consumption of the chain scenario.

The average sensor energy consumption in the realistic scenario is shown in Fig.15. The trend of curves

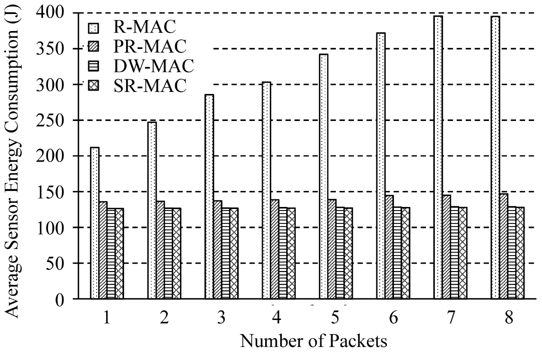


Fig.15. Average sensor energy consumption of the realistic scenario.

is similar to Fig.15. R-MAC consumes most energy in all protocols and its energy consumption of SR-MAC is least.

To schedule the data packets to forward on the multi-hop path, just like R-MAC, SR-MAC has to carry on some routing information in the SRF. Therefore, SR-MAC consumes more energy to transmit a single SRF than S-MAC does to transmit a single RTS/CTS. On the other hand, sensors in SR-MAC have to listen during the whole DATA period so they consume more energy. However, our simulation results show that SR-MAC consumes least energy. In addition, the simulation in [10] shows that R-MAC has higher energy efficiency than S-MAC. The major reason is that R-MAC transmits less control frames than S-MAC does. Less transmission means less receiving or overhearing, which decreases the total energy consumption of R-MAC. Another reason is that the data packets can be moved away from the busy area as soon as possible in R-MAC by forwarding the packets over multi-hop in a single cycle, which helps to reduce the contention in this area. Therefore, it can reduce the energy waste of packet retransmission because of the collision.

4.6 Throughput Evaluation

Although network throughput is not a crucial metric in MAC protocol design for WSNs, it is still important when the traffic comes in a burst. Therefore, in this subsection, we evaluate the network throughput of SR-MAC. We vary the traffic loads by varying the number of the packets generated by an event on a single node.

In both scenarios, the number of packets generated by an event varies from 1 to 8. In the chain scenario, CBR generates an event every 20 s, so the input rate of the source node varies from 1 packet per 20 s (pkts/20 s) to 8 pkts/20 s. In the realistic scenario, RCE still generates an event every 200 s. We assume that the sensing radius is 200 m in the throughput simulation. Based on the parameters in Table 4, there are averagely 10.6 nodes to detect the even simultaneously. Therefore the average input rate of the realistic scenario varies from (10.6×1) pkts/200 s to (10.6×8) pkts/20 s.

Fig.16 gives the simulation results of the chain scenario. For DW-MAC, after $N_{\text{pkt}} = 5$, the throughput begins to be lower than the input rate and almost keeps no change. R-MAC's throughput is lower than the input rate and almost keeps no change when $N_{\text{pkt}} = 3$. However, the throughputs of PR-MAC and SR-MAC are nearly equal to the input rate of the network.

The network throughput in the realistic scenario is shown in Fig.17. The trends of curves are similar to those of Fig.16. The throughput of R-MAC is a little less than SR-MAC when $N_{\text{pkt}} \leqslant 2$ and drops down obviously when $N_{\text{pkt}} \geqslant 3$. R-MAC's lowest throughput is only about 17% of the input rate. DW-MAC's throughput drops down obviously when $N_{\text{pkt}} = 4$ and is only 60% of the input rate when $N_{\text{pkt}} = 8$. When $N_{\text{pkt}} \leqslant 5$, PR-MAC's throughput is the same as SR-MAC's. However, SR-MAC's throughput outperforms PR-MAC's when $N_{\text{pkt}} \geqslant 6$. Furthermore, the throughput of SR-MAC is a little less than the input rate and hardly effected by the input rate.

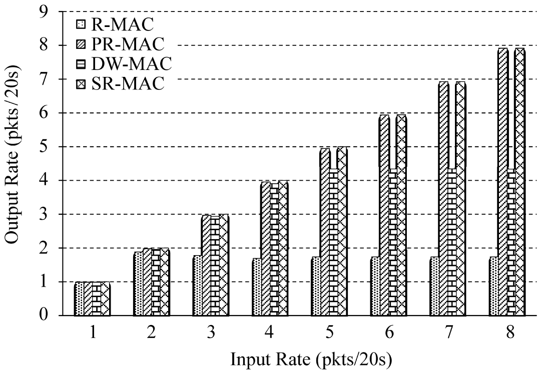


Fig.16. Throughput of the chain scenario.

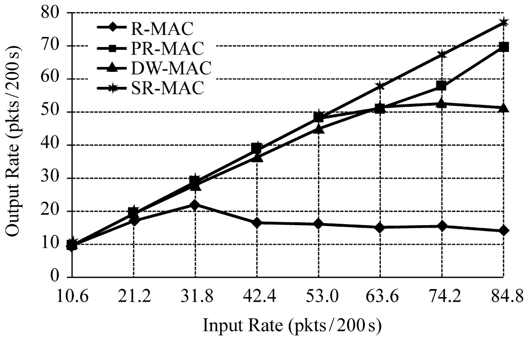


Fig.17. Throughput of the realistic scenario.

4.7 Network Density Evaluation

With the network deployment density increasing, the neighbors of each node increase heavily and more nodes detect the event simultaneously. Consequently, more nodes contend the channel during the DATA period and collide with each other, which may result in the performance degrade of the protocols. In this subsection, we evaluate the network density's effect on the performance of SR-MAC.

In our simulation, we only use the realistic scenario to evaluate the performance of the protocols with different network densities. We keep the radio's transmission range of 250 m, and we change the network density by varying the total number of nodes in the 1 000 m by 1 000 m square area. Table 5 gives the average number of neighbors of each node with different network densities.

Table 5. Average Number of Neighbors of Each Node

Number of nodes	80	90	100	110	120
Average number of neighbors	12.7	14.3	15.9	17.5	19.1

Fig.18 shows the EDL, EDR and average sensor energy consumption of all protocols with different network densities. From the simulation results, we find that the network density has the lightest effect on SR-MAC's performance. For example, SR-MAC has the smallest rate of EDL with network density increasing, and its EDR always keep 100%.

5 Conclusions

In this paper, for the event detection applications in WSNs, we presented a new slot-reserved, low-latency, multiple packets transmission and synchronous duty-cycle MAC protocol, called as SR-MAC. SR-MAC divides the DATA period into data slots and the SLEEP period into frames. Furthermore, each frame in the SLEEP period is divided into sleep slots. In order to schedule nodes to deliver multiple packets generated by an event on a single node, SR-MAC assigns a frame-slot pair of the SLEEP period for each node with pending data. By this mechanism, SR-MAC can forward multiple packets on the multi-hop path in one operational cycle without collision.

Through detailed ns-2 simulation, we compared SR-MAC with R-MAC, DW-MAC and PR-MAC. The simulation results show that SR-MAC outperforms these protocols, with lower EDL, higher EDR, higher network throughput and guaranteed energy efficiency. Especially, under ultra-heavy traffic loads, SR-MAC reduces EDL by about 60% compared with PR-MAC, and maintains 100% EDR with less energy consumption. Even though the network is deployed very densely, SR-MAC still has excellent performance.

Despite of all the advantages of SR-MAC, there are still some issues left open for future research. Firstly, in order to forward data packet on the multi-hop path in a single cycle, all nodes have to listen the wireless channel during the entire DATA period, which wastes significant energy. The nodes without communication may sleep as soon as possible. Secondly, We only evaluate the performance of SR-MAC with software simulation. We hope to implement SR-MAC on a realistic sensor network platform, such as TinyOS.

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, New York, NY, USA, 1997.

[2] Stemm M, Katz R H. Measuring and reducing energy consumption of network interfaces in hand-held devices. *IEICE Trans. Communications*, 1997, E80-B(8): 1125-1131.

[3] Polastr J, Hill J, Culler D. Versatile low power media access for wireless sensor networks. In *Proc. the 2nd Int. Conf. Embedded Networked Sensor Systems*, Nov. 2004, pp.95-107.

[4] Ye W, Heidemann J, Estrin D. An energy-efficient MAC protocol for wireless sensor networks. In *Proc. the 21st Annual Joint Conference of the IEEE Computer and Communications Societies*, Jun. 2002, pp.1567-1576.

[5] Du S, Saha A K, Johnson D B. RMAC: Routing-enhanced duty-cycle MAC protocol for wireless sensor networks. In *Proc. the 26th IEEE Conf. Computer Communications*, May 2007, pp.1478-1486.

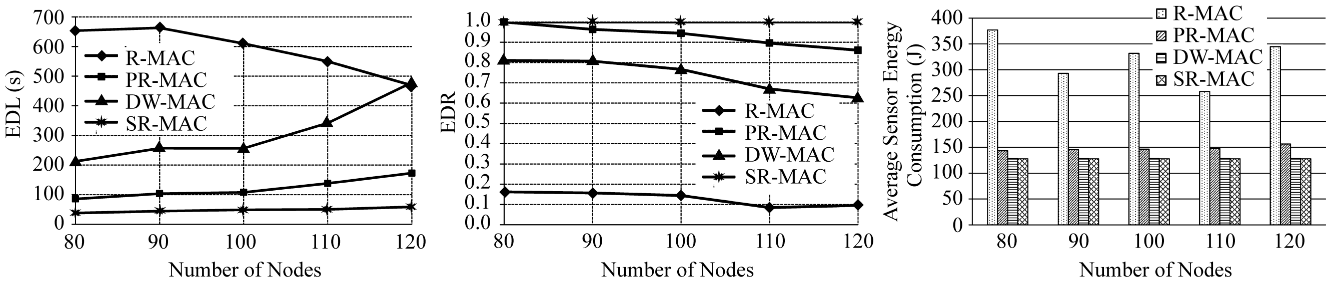
[6] Sun Y, Du S, Gurewitz O, Johnson D B. DW-MAC: A low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks. In *Proc. the 9th Int. Symp. Mobile Ad Hoc Networking and Computing*, May 2008, pp.53-62.

[7] Canli T, Khokhar A. PRMAC: Pipelined routing enhanced MAC protocol for wireless sensor networks. In *Proc. the Int. Conf. Communication*, Jun. 2009, pp.1-5.

[8] Buettner M, Yee G V, Anderson E, Han R. X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proc. the 4th International Conference on Embedded Networked Sensor Systems*, Oct. 31-Nov. 3, 2006, pp.307-320.

[9] Sun Y, Gurewitz O, Johnson D B. RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In *Proc. the 6th Conf. Embedded Network Sensor Systems*, Nov. 2008, pp.1-14.

[10] Tang L, Sun Y, Gurewitz O, Johnson D B. PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks. In *Proc. the 30th IEEE Int. Conf. Computer Communications*, Apr. 2011, pp.1305-1313.



(a) (b) (c)

Fig.18. Network density evaluation. (a) EDL. (b) EDR. (c) Average sensor energy consumption.

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

[12] van Dam T, Langendoe K. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proc. the 1st Int. Conf. Embedded Networked Sensor Systems*, Nov. 2003, pp.171-180.

[13] El-Hoiydi A, Decotignie J D. WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks. In *Proc. the 1st International Workshop on Algorithmic Aspects of Wireless Sensor Networks*, Jul. 2004, pp.18-31.

[14] Lu G, Krishnamachari B, Raghavendra C S. An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks. In *Proc. the 18th Int. Parallel and Distributed Processing Symposium*, Apr. 2004, pp.224-231.

[15] Lu G, Sadagopan N, Krishnamachari B, Goel A. Delay efficient sleep scheduling in wireless sensor networks. In *Proc. the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Mar. 2005, pp.2470-2481.

[16] Keshavarzian A, Lee H, Venkatraman L. Wakeup scheduling in wireless sensor networks. In *Proc. the 7th ACM Int. Symp. Mobile Ad Hoc Networking and Computing*, May 2006, pp.322-333.

[17] Elson J, Girod L, Estrin D. Fine-grained network time synchronization using reference broadcasts. In *Proc. the 5th Symp. Operating Systems Design and Implementation*, Dec. 2002, pp.147-163.

[18] Ganeriwal S, Kumar R, Srivastava M B. Timing-sync protocol for sensor networks. In *Proc. the 1st Int. Conf. Embedded Networked Sensor Systems*, Nov. 2003, pp.138-149.

[19] Ye W, Silva F, Heidemann J. Ultra-low duty cycle MAC with scheduled channel polling. In *Proc. the 4th Int. Conf. Embedded Networked Sensor Systems*, Oct. 31-Nov. 3, 2006, pp.321-334.

[20] Shnayder V, Hempstead M, Chen B, Allen G W, Welsh M. Simulating the power consumption of large-scale sensor network applications. In *Proc. the 2nd Int. Conf. Embedded Networked Sensor Systems*, Nov. 2003, pp.188-200.

[21] Hull B, Jamieson K, Balakrishnan H. Mitigating congestion in wireless sensor networks. In *Proc. the 2nd Int. Conf. Embedded Networked Sensor Systems*, Nov. 2004, pp.134-147.

[22] Cano C, Bellalta B, Barceló J, Sfairopoulou A. A novel MAC protocol for event-based wireless sensor networks: Improving the collective QoS. In *Proc. the 7th Int. Conf. Wired/Wireless Internet Communications*, May 2009, pp.1-12.

Hong-Wei Tang received the B.S. and M.S. degrees in computer science and technology from the College of Computer Science at National University of Defense Technology, Changsha, in 2001 and 2003, respectively. He is currently a Ph.D. candidate of School of Computer at National University of Defense Technology. His research area includes parallel and distributed computing, network protocols, wireless sensor networks.

Jian-Nong Cao is a full professor and acting head of the Department of Computing at Hong Kong Polytechnic University. He is also the director of the Internet and Mobile Computing Lab in the department. His research interests include computer networks, mobile and pervasive computing, parallel and distributed computing, fault tolerance, and middleware. He is the Coordinator in Asia of Technical Committee on Distributed Computing (TPDC), IEEE Computer Society, a senior member of IEEE, a member of ACM (SIG-Mobile), a senior member of CCF and a member of CCF's Technical Committee on System Software and Technical Committee of Pervasive Computing.

Xue-Feng Liu received the B.S. degree in automatic control and the M.S. degree in control engineering from Beijing Institute of Technology in 1999 and 2002 respectively. He worked as a software engineer in Beijing Institute of Control Engineering from 2002 to 2003. He got his Ph.D. degree in aerospace engineering at Department of Aerospace Engineering, University of Bristol in 2008. He currently is a postdoctoral research fellow in the Department of Computing at the Hong Kong Polytechnic University. His research interests include wireless sensor networks, structural healthy monitoring, and cyber-physical systems.

Cai-Xia Sun received the M.S. and Ph.D. degrees in microelectronics from College of Computer Science at National University of Defense Technology in 2001 and 2006, respectively. She is currently an associate professor of School of Computer at National University of Defense Technology. Her research interests include parallel and distributed computing, fault tolerance and high performance computing.