Applied Mathematics & Information Sciences
An International Journal

RUF-MAC: Related and Urgent First MAC for Wireless Sensor Networks

Long Chen[1]*, Xiao-jun Li*[2]*, Yuan-bo Guo*[3]*, Liang-min Wang*[2,*]

[1]Institute of Automobile Engineering, Jiangsu University, Zhenjiang, 202013, P. R. China
[2]Department of Computer Science, Jiangsu University, Zhenjiang, 212013, P. R. China
[3]School of Electronic Technology, PLA Information Engineering University, Zhengzhou, 450004, P. R. China

Received: 11 Sep. 2012, Revised: 21 Dec. 2012, Accepted: 29 Jan. 2013

Abstract: Medium Access Control (MAC) protocols of Wireless Sensor Network (WSN) are often required to have special characteristics in concrete applications. Related and Urgent message should be sent First (RUF) is a typical scenario in many applications of WSN. In this paper, we focus on presenting a MAC protocol for RUF applications in WSN. First of all, through analysis, some concrete cases of RUF scenario are listed, from which 4 common requirements of MAC protocols are summed up. Then a novel RUF-MAC is devised to meet these requirements. By modifying the frame structure of IEEE 802.15.4 standards, new algorithms are presented for the assignment and adjustment of time slots, of which some measures are introduced into RUF-MAC, such as sub-frames competition, priority assignment and adaptive period adjustment. The simulation results of data from RUF scenario show that the performance of RUF-MAC is better than that of other MAC protocols for WSN in RUF scenario.

Keywords: Wireless sensor networks, media access control protocols, energy management

1. Introduction

In the architecture of wireless sensor networks, Medium Access Control (MAC) protocol determines the allocation of the wireless channel. Responsible for assigning wireless communication resources, the energy efficiency quality of MAC will directly have influence on the performance of network.

Most previous studies try to save energy to prolong the life of the networks for energy saving is an important aspect of MAC protocol. But there are some other important aspects for different applications. For example, real-time reporting of accidents is more important than energy saving in App-MAC [1]. In addition, in many cases some information should take priority of others in WSN applications. Now we introduce some typical examples about RUF applications with multimodality wireless sensor network applications.

Case 1: Intelligent House-holding System. WSN is used to monitor and integrate electronic subsystem such as electronic cooking device, automatic lighting subsystem and security guarding subsystem. This application system has some special characteristics. Firstly, some events are happened by accident. For instance, electronic device is broken or fire alarm is ringing. Secondly, some emergency messages should be sent in the priority order. Thirdly, some decisions or judgments are deduced from related messages. For example, a message shows that the gas burner is open, but another message reports that the temperature in the cooking room isn't changed. Maybe it indicates a gas leakage. But the system can't find the problem if it has no one of the two messages. That is to say, the two related messages sensed by different nodes should be sent together.

Case 2: Traffic Reminding System for Bidirectional Corner. WSN can sever as a traffic reminder in a bidirectional corner to avoid the collision of two cars. Two cars may be come into collision because each driver can't see the other in another corner. WSN based traffic reminding system can sense the events in the other corner and remind the driver as enlarging his vision. This reminding system has the RUF scenario either. In this case, the traffic collision happens by accident, the potential collision is more urgent than other thing and two related messages from different location about a collision should be sent to the sink together.

---

[*] Corresponding author e-mail: wanglm@acm.org

From the above cases, a MAC protocol for RUF scenario should have 4 characteristics: no fixed schedule for changeable event frequency in inactive period, low latency and prioritized channel competition for different events, fair transmissions of messages related to urgent events and energy saving. In this paper, we present a novel IEEE 802.15.4 based MAC to meet these requirements. Comparing with the other hybrid MAC, such as App-MAC [1], Z-MAC [2], PW-MAC [3] and Crankshaft [4], our contributions are two-fold: first, we conclude a kind of common scenarios and its requirement for MAC protocol. Second, an applied MAC which is suitable for the scenarios is proposed by us. In comparison with the most relevant work App-MAC, we make some improvements as follows: (1) The related message can be sent first by raising its Priority; (2) The length of CAP, CFP and inactive period are tunable according to the number and status of events.

The rest of the paper is organized as follows: In section 2, point out why typical WSN MAC protocols are not applicable for RUF scenario and present our RUF-MAC to meet the requirements of the RUF scenarios; Some fundamental structures of our RUF-MAC are given in section 3; Based on the new structures, the CAP assignment algorithm and time slots adjustment algorithm are presented in section 4. Performances comparison of our RUF-MAC, Z-MAC and App-MAC on delivery latency, event fairness, and the efficiency are presented in section 5. We conclude the paper in section 6.

2. Related Work

MAC protocols are often directly linked to the overall performance of WSNs. There are lots of MAC protocols devised by researchers. However, different applications require different MAC protocols. According to the channel access strategies, we classify these MAC protocols into 3 types: contention-based protocols, schedule-based protocols and hybrid MAC protocols.

2.1. Contention-based MAC protocols

S-MAC [5], T-MAC [6], B-MAC [7] and Sift [8] are typical contention-based MAC protocols. They access wireless channel through a contention mode, which decides whether retransmitting or discarding packages when collision happens. Usually, global information of network is not required in contention-based MAC protocols, so they are easy to implement and have good scalability. But in the meantime, their energy cost are high [9] and cause serious packet collisions in high frequency applications, which in return lead to high transmission delay and high energy consumption.

2.2. Schedule-based MAC protocols

Such as traffic-adaptive MAC [10] (TRAMA) and TDMA-wakeup MAC [11] (TDMA-W), each nodes access channel in a given time slot by a mapping algorithm in Schedule-based MAC protocols. In these protocols, nodes communicate with each other through a prearranged schedule and neighbor nodes transmit data in different time slots to avoid packet collisions [9]. The nodes turn on their wireless transceivers in scheduled time slots, or else, they will turn the power off to reduce energy consuming from idle listening. These measures decrease two main energy consumption sources in wireless sensor networks, namely idle listening and packet collisions [12]. Schedule-based MAC allocates the time slots according a synchronized clock, which is a difficult problem in sensor networks. The scalability of them will be bad when the network scale is larger.

2.3. Hybrid MAC protocols

Z-MAC is a hybrid MAC based on TDMA and CSMA. TDMA is the traditional schedule-based MAC protocol, and CSMA is an earliest contention-based MAC protocol. Combining the goodness of schedule-based and contention-based MACs and offsetting their weakness, Z-MAC becomes robust to synchronization errors, slot assignment failures, and time-varying channel conditions. It can achieve high channel utilization under high contention and reduce collision among two-hop neighbors with a low cost. Unfortunately, Z-MAC is not suitable for RUF scenario. One reason is that Z-MAC's performance always falls back to that of CSMA under low contention [2]. But there is no contention in long idle listening when no accident happens in RUF applications. Moreover, In order to give fairness to all nodes, each node in Z-MAC has an assigned time slot and has highest priority to transmit its data in its own slot. When other nodes are sending some unimportant messages, the node has urgent messages should wait for the slots of itself.

App-MAC [1] supports prior event delivery, in which the nodes with high priority messages can occupy the slots other nodes. However, App-MAC fixes the lengths of time slots as Contention time Slots (CS), Reservation time Slots (RS) and inactive period, which may not suitable for the mass messages of accidents in RUF scenarios. In addition, App-MAC does not consider fairness of the correlated aspects of an event. In RUF applications, the sink should combine data in different nodes to judge whether an urgent event happens. Thus, after sending one message for a possible urgent event, the related message should be sent in real time by stealing other nodes' slots.

PW-MAC [3] is a recently presented hybrid protocol, in which nodes are waked up to receive at randomized, asynchronous times. PW-MAC minimizes sensor node energy consumption by enabling senders to predict receiver wakeup times. In order to get accurate predictions, PW-MAC introduces an on-demand prediction error correction mechanism that effectively addresses timing challenges such as unpredictable hardware and operating system delays and clock drift [3]. Focusing on energy efficiency under multiple concurrent multihop traffic flows and under hidden-terminal scenarios and scenarios in which nodes have wakeup schedule conflicts, PW-MAC significantly outperformed other protocols. Obviously, PW-MAC is not presented for our RUF scenario without consideration of events priority and related messages.

3. Fundamental Structure of RUF-MAC

We follow the standard of IEEE802.15.4 [13] and adopt the beacon-enabled mode to design super-frame of RUF-MAC. Under fully considering the special requirements of RUF scenario, we design the structures of superframe and bacon packets.

3.1. Superframe structure

Superframe of RUF-MAC is based on IEEE 802.15.4, which begins with a beacon period, and followed with an active period and an inactive period. The structure of our super-frame is shown in Fig.1. Similar to App-MAC [1], it has four parts: beacon period (Beacon), Contention Access Period (CAP), Contention Free Period (CFP) and inactive period (Inactive). In beacon period, the sink node serves as the coordinator broadcasting beacon packet periodically to acknowledge packet transmission and synchronize the sensor nodes. In CFP period, sensor nodes transmit data to the sink in their own time slots allocated by the sink, while all nodes are turned off to save energy in inactive period.

There are two differences between RUF-MAC and App-MAC. Firstly, considering the fairness of the messages in an urgent event, we divide the CAP period into several subframes which are provided to nodes according to their related events. In RUF-MAC, the node competes for the slots not only by its event priority but also the event type. Namely, if two nodes compete for the time slots in a subframe with the same priority, the node that sends a message related to the event of this subframe (it is signed by Group ID as shown in the following) will success. This measure provides more chances for related messages and makes the later messages have fairness with the previous message in transmitting. Secondly, the lengths of CFP and inactive period are tunable in RUF-MAC. Different from App-MAC, RUF-MAC can reduce the idle listening and prolong inactive period to save energy when there are no events. The dynamically tunable lengths not only save the energy, but also permit to add active time slots in high event frequency, which is very suitable for RUF scenario.

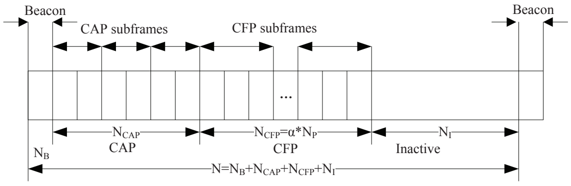

Figure 1 Superframe structure of RUF-MAC

3.2. Beacon packet

The beacon packet plays an important role in synchronization, packet acknowledgement and time slots allocation. For reducing the packet collision and save energy, we employ the beacon packet to reduce the number of control packet. Combing with the structure of Active Message in TinyOS [14], we define the structure of beacon packet as shown in Fig.2.

In beacon packet, the first field is the ID of beacon produced by the system time, which is used in two occasions. One is serving as a time stamp to synchronize sensor nodes, the other is preventing the packet mismatching of sensors caused by beacon packet missing. The following field is packet acknowledgement bitmap, which is corresponding to CFP subframe. Each bit of packet acknowledgement bitmap indicates packet acknowledgement in each CFP subframe. If the sink node receives data packet successfully, then the corresponding bit is 1, or it will be 0. The third field stores the length of CAP assignment list and CFP assignment list. The length of assignment list is related with the number of reported events. If there are lots of events to report, the list will be long, or else, the list will be short.

The residual fields are the CAP and CFP assignment lists. Each assignment entry of the list describes one subframe of CAP or CFP. CAP assignment list (CAP-Assign-List) is corresponding to CAP subframes. Each CAP assignment entry contains the description of the subframe, including the group ID of nodes (Group-ID), the sensor type (Sensor-Type), the event priority (Event-Priority) and the number of time slots (Slot-Num). CFP assignment list (CFP-Assign-List) is corresponding to CFP subframes. Each CFP assignment entry contains the description of the subframe, including the nodes ID (Node-ID), an event ID (Event-ID) and the number of time slots (Slot-Num).

A difference of RUF-MAC between App-MAC is that RUF-MAC has a Group-ID field in each CAP assign list. Group-ID is used to describe the group of nodes, which is correlated with a given event and the CAP subframe to provide fairness for messages related to the events.

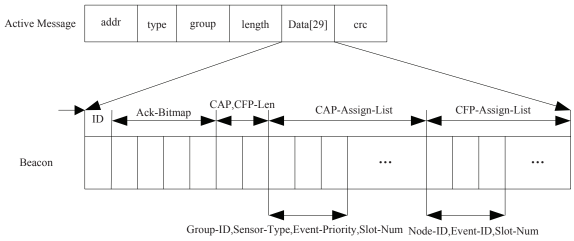

Figure 2 Beacon packet in RUF-MAC

4. Algorithms for Time Slots

In this section, we present a CAP assignment algorithm to reduce packet collisions. Then, by allocating time slots in active and inactive period, we present an adjustment algorithm to reduce idle listening and save energy.

4.1. CAP assignment algorithm

Packet collisions, retransmission and retransmission delay have high influence on delivery latency and energy consumption [15]. CAP assignment algorithm reduces the packet collisions and improves the channel utilization by allocating different nodes into different number of time slots.

CAP assignment algorithm is performed by the sink. We denote that a sink node has N_G (Number of Groups) event lists to store information of events reported from relative nodes. The information includes priorities events, the number of packets and data of the reporting node. In each list, the events are ordered by their priority. The priorities of evens and node type are set by the sink for the nodes to compete the time slots of CAP.

In system model, each node is sorted into a group G_i (i =1, ..., N_G). Any nodes in G_i take charge of a given event detection from different aspect and different regions. We use L_i (i =1, ..., N_G) to store the event information detected by nodes in group G_i (i =1, ..., N_G). So there are N_G subframes in CAP of the superframe structure. We use CS_i (i =1, ..., N_G) to denote CAP subframes related to the given event reported by the nodes from G_i.

Taking the initial event list L_i and the number of event lists N_G as the input, the CAP time slots assignment algorithm outputs time slots assignment information CS [N_G] of CAP. The pseudo-code of the CAP assignment algorithm is shown in Fig.3.

In initial phase of the algorithm, all lists are null. The sink node sets CAP subframe CS_i (i =1, ..., N_G) which can be competed by nodes from group G_i (i =1, ..., N_G). Then the sink node writes the assignment of time slots into beacon packet. When the sink node receives some reported event from the node in G_i, it will record the event information into the corresponding list L_i. When the list is not null, the events compete for the CAP subframe by their priority. If these is a null list L_j (j =1, ..., N_G), the event related to CS_j will be set with the lowest priority.

Algorithm 1: CAP Time Slots Assignment

```
Function CAP-Assign
Begin function
    for $i$ from 0 to $N_G$ do
        Sort($L_i$)    // quick sorting algorithm
    end for
    for $i$ from 0 to $N_G$ do
        if($L_i$! = NULL)
            CS[$i$] $\rightarrow$ pri = $L_i$ $\rightarrow$ head.pri
        else
            CS[$i$] $\rightarrow$ pri = -1
            CS[$i$] $\rightarrow$ group = $i$
        end if
    end for
    return CS
End Function
```

Figure 3 Pseudo-code of the Algorithm for CAP Time Slots Assignment

The complexity of Algorithm 1 includes event lists sorting and event priority setting. A quick sorting algorithm is used for sorting N_G lists here, whose complexity is $O(nlogn)$. The complexity of event priority setting is $O(N_G)$. So the algorithm complexity of algorithm 1 is $O(nlogn)$+$O(N_G)$, where n is the length of event list.

4.2. Time slots adjustment algorithm

The idle listening of nodes without communication task is the main source of energy consumption. In RUF scenario, the event happens by accident. Therefore, in the low event frequency, there will be a long time without communication. We present the time slots adjustment algorithm to prolong the inactive period and reduce energy consumption.

According to the number of reported events and packets waiting to be sent, we adjust the time slots assignment in superframe dynamically. When the events or packets are less, we reduce time slots in CFP while increasing time slots in inactive period. In this way, the idle listening will be reduced. Thus, the energy consumption of nodes will be reduced and the life cycle of the network will be extended.

We denote the N as the number of time slots, N_B, N_I as the number of time slots in Beacon and inactive periods respectively, as well as, N_{CAP}, N_{CFP} as the number of time slots in CAP and CFP respectively. Similar to App-MAC, we set N and N_B as the constants while N_{CAP}, N_{CFP} and N_I are variables. We denote N_P and α as the number of packets waiting for transmission and the number of time slots required to send a packet respectively. Then, it holds

```
Algorithm 2: Time Slots Adjustment
Function time-slots Adjust
Begin Function
    for $i$ from 0 to $N_G$ do
        $PacketNum += GetPacketNum(L_i)$
    end for
    $N_{CFP} = \alpha \times PacketNum$
    $N_I = N - N_B - N_{CAP} - N_{CFP}$
End Function
```

Figure 4 Pseudo-code of the adjustment algorithm for time slots

that:

$$N_B + N_{CAP} + N_{CFP} + N_I = N \tag{1}$$

$$N_{CFP} = \alpha N_P \tag{2}$$

From Equation(2), we can see that the number of time slots in CFP is related with the number of waiting packets. Less time slots are required when the waiting packets are less. Then we can reduce the length of CFP and increase the sleep cycle of nodes to achieve the goal.

Combining Equation(1) and Equation(2), we have

$$N_I = N - N_B - N_{CAP} - \alpha N_P \tag{3}$$

Obviously, the length of inactive period will increase with the decrease of CFP, which means that, when there are less communication tasks, the nodes can sleep to reduce energy consumption. For example, if $N = 30$, $N_B = 2$, $N_{CAP} = 7$ and the initial value $N_{CFP} = 16$, $N_I = 5$, $\alpha = 0.5$, $N_P = 20$, then we can get N_{CFP} and N_I as follows:

$$N_{CFP} = \alpha \times N_P = 0.5 \times 20 = 10 \tag{4}$$

$$N_I = N - N_B - N_{CAP} - \alpha N_P = 11 \tag{5}$$

In this case, we add 6 time slots to N_I, which means that all nodes will reduce 6 time slots for idle listening. So the energy consumption of nodes will decrease significantly.

Takes event list L_i and time slots assignment information as input, the pseudo-code of our time slots adjustment algorithm is shown in Fig.4. The complexity of algorithm 2 includes the statistics of the packets and value assignment of time slots. But the value assignment of time slots is out of the loop body. So the complexity of algorithm 2 is $O(N_G)$.

5. Experimental Result

In this section, we use nesC language [16] to implement RUF-MAC on TOSSIM simulator, a platform based on TinyOS. Then we make a comparison of performances between App-MAC, Z-MAC and our RUF-MAC.

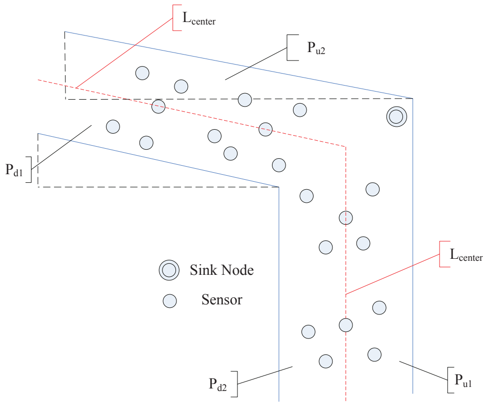

Figure 5 Traffic reminding system for the entrance of Underground Car Park

5.1. Simulation parameters

We select 20 Micaz motes nodes and mib520 Micaz mote node powered by two 1.5V AAA batteries for construction of the network, where mib520 is use as sink. The sink connects to a USB interface on PC by wire.

The simulation scenario was selected from case 2, a RUF application mentioned in section 1. As shown in Fig.5, there was a traffic reminding system at the entrance of an underground car park, where several types of wireless sensors, such as vibratory, sound and infrared sensors are deployed to monitor the cars, persons and other things. The network transmits the sensed data to the sink which will fuse the data and decide whether to give a traffic warning. In the case shown in Fig.5, an urgent collision is related to two events: one is a car entering the park at P_{d1}, and the other is a car getting out of the park from P_{u1}. Only when the sink gets the two related messages can it make a decision, while the response time is determined by the later coming message.

5.2. Performances of RUF-MAC

In this subsection, we make a comparison of performances between App-MAC, Z-MAC and our RUF-MAC containing event delivery latency, event fairness, and energy consumption.

5.2.1. Event delivery latency

App-MAC [1] is defined as the latency of event delivery, a time period, from the time when the event happens to that when the sink receives all the data. So the event delivery latency is related with many factors, such as channel allocation algorithm, event frequency, link quality and so on.

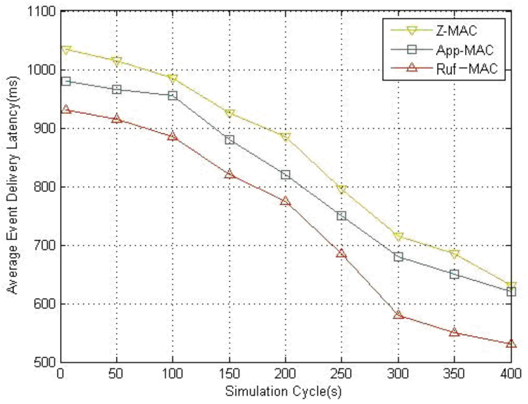

Figure 6 All events with high priority

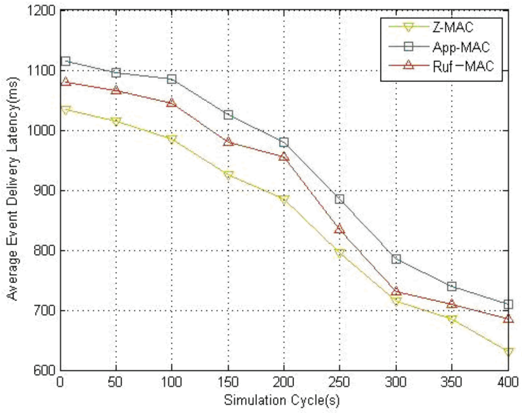

Figure 7 All events with low priority

Considering events with two different priorities, we compute and compare the event delivery latency of each priority with Z-MAC and App-MAC in the simulation. There are 20 events (10 with high priority and 10 with low priority) happen randomly in a cycle which changes from 5s to 400s.

In Fig.6, the event delivery latency of RUF-MAC and App-MAC are smaller than that of other Z-MAC because CAP assignment algorithm allocates time slots for high priority events first which limits the data transmission of low priority events in RUF-MAC and App-MAC. That is to say, the events with high priority occupy the transmitting chance of the events with low priority.

In Fig.7, the performance of Z-MAC is better than RUF-MAC and App-MAC because RUF-MAC and App-MAC sacrifice the chances of events with low priority to ensure the real-time transmitting of high priority events.

Combining Fig.6 and Fig.7, it is clear that our RUF-MAC shows better performance than App-MAC in sending events of both high and low priority. Because events should be transmitted just in their own time slots in App-MAC, while higher prioritized events can steal time slots of lower prioritized events in RUF-MAC.

5.2.2. Event fairness

The traditional definition of fairness focuses on the amount of bandwidth shared by nodes. Du et al [1] point out that the allocation of the bandwidth should be related with event delivery latency, and they define event fairness index as follows:

$$I_e = \frac{1}{n}\sum_{p=1}^{n}\left(\frac{1}{n_p}\sum_{i=1}^{n_p}\sqrt{(L_{p_i}-\overline{L_p})^2}\right) \qquad (6)$$

Where n is the priority number, L_{p_i} is the event data delivery latency of the i^{th} event with priority p, $\overline{L_p}$ and n_p are the average event delivery latency and the total number of events with priority p respectively. The event fairness index indicates the average standard deviation of event delivery latency for all priorities. The value of the sensor fairness index will be small when MAC protocol treats the same prioritized events fairly [1].

Fig.8 shows the variation of the event fairness index with simulation cycle. The event fairness index of Z-MAC becomes smaller seriously when simulation cycle increases longer, but App-MAC and RUF-MAC doesn't change obviously. The reason is that Z-MAC can't treat all the events fairly in high contention model. Especially, there maybe two or more messages are correlated to an event in our RUF scenario. That is also a reason why RUF-MAC performs a little better than App-MAC. Meanwhile, RUF-MAC divides CAP into several subframes according to the priority of events in which different events with the same priority will compete for different subframes. This measure provides more fairness to different events with same priority than that in App-MAC.

5.2.3. Energy consumption

Powered by the battery, the energy of sensor node is very limited, therefore, energy saving is a very factor for sensor node designing. We evaluate MAC's performance in energy saving according to a quotient which is calculated through the total energy consumption divided by the number of transmitted packets. It is clear that, the quotient is the average energy consumption of transmitting a packet and the energy consumption of MACs increases with the length of simulation cycle because the energy cost from idle listening.

From Fig.9, it is clear that the energy consumption of Z-MAC is the lowest when the simulation cycle is small while when the simulation cycle is small it becomes the

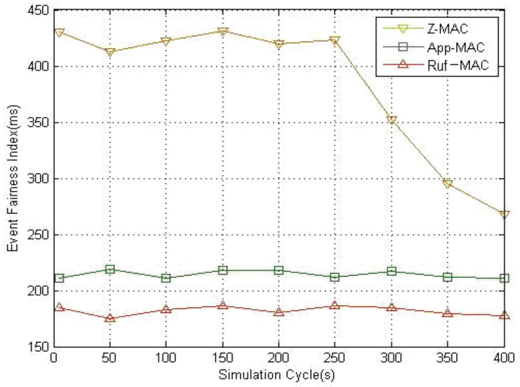

Figure 8 Event fairness index

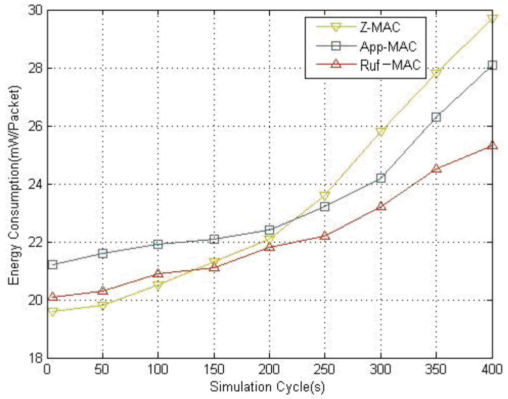

Figure 9 Energy consumption efficiency

highest. However, the energy consumption of RUF-MAC is the lowest when the simulation cycle is long, because, by using adjustment algorithm, it dynamically adjusts the number of time slots in inactive period according to the number of events. In RUF applications, the number of events changes by accident. With the reduction of packets, RUF-MAC prolongs the inactive period to reduce the idle listening. As a result, our RUF-MAC performs better in RUF scenarios.

6. Conclusion

We present a RUF-MAC protocol to meet these special requirements in RUF scenario. In the protocol, we set a tunable length of inactive period and present an adjustment algorithm to reduce the idle listening time when there are no too many events to report, and increase the time slots in CFP period when suddenly mass messages are need to send. Moreover, we set messages related to different events with different priorities. Then the urgent messages will have higher priorities and occupy more chance to be transmitted in CAP. Further, we divide the CAP period into some subframes in superframe structure, and construct the relations between the subframes and the events. Thus the related messages to be sent to sink will have higher event fairness. The simulation results demonstrate that RUF-MAC performs well in the RUF scenario.

Acknowledgement

The work of this paper is supported by National Natural Science Foundation of China (61272074), and Natural Science Foundation of Jiangsu Province (BK2011464), and the author Liang-min WANG is supported by “Qing Lan Project” for young and middle-aged academic leader of Jiangsu Province.

References

[1] J.Z Du, W.S Shi, App-MAC: An Application-Aware Event-Orient MAC Protocol for Multimodality Wireless Sensor Networks. IEEE Transactions on Vehicular Technology, **57**, 3723–3731 (2008).

[2] I. Rhee, A. Warrier, M. Aia, J. Min and M.L Sichitiu, Z-MAC: A Hybrid MAC for Wireless Sensor Networks. IEEE/ACM Transactions on Networking, **16**, 511–524 (2008).

[3] L Tang, Y Sun, O Gurewitz, and D Johnson, PW-MAC: An Energy-Efficient Predictive-Wakeup MAC Protocol for Wireless Sensor Networks. In Proceedings of the 30th IEEE International Conference on Computer Communications, **13**, 1305-1313 (2011).

[4] G.P. Halkes and K.G. Langendoen, Crankshaft: An energy-efficient MAC-protocol for dense wireless sensor networks. In Proceedings of the 4th European Conference on Wireless Sensor Networks, **2**, 228-244 (2007).

[5] W. Ye, J. Heidemann and D. Estrin, Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks. IEEE/ACM Transactions on Networking, **12**, 493–506 (2004).

[6] T.van Dam, K. Langendoen, An Adaptive Energy-efficient MAC Protocol for Wireless Sensor Networks. In Proceedings of the 1st ACM Conference on Embedded Networked Sensor System, **1**, 171–180 (2003).

[7] J. Polastre, J. Hill and D. Culler, Versatile Low Power Media Access for Wireless Sensor Networks. Proceedings of the Second International Conference on Embedded Networked Sensor System, **1**, 95–107 (2004).

[8] K. Jamieson, H. Balakrishnan and Y.C. Tay, Sift: MAC Protocol for Event-driven Wireless Sensor Networks. EWSN, LNCS, **3868**, 260–275 (2006).

[9] H.G Gong, M Liu, C.Y Yu, X.M Wang and L.J Chen, Energy-Efficient Event Driven TDMA Protocol for Wireless Sensor Network. ACTA ELETRONICA SINICA, **35**, 1843–1848 (2007).

[10] V. Rajendran, K. Obraczka and J.J.Garcia, Energy-Efficient, Collision-free Medium Access Control for Wireless Sensor Network, In Proceedings Of the ACM SenSys, **1**, 181–192 (2003).

[11] Z.H Chen, A. Khokhar, Self Organization and Energy Efficient TDMA MAC Protocol by Waking Up for Wireless Sensor Networks, Proceedings of the Sensor and Ad Hoc Communications and Networks. IEEE Computer Society, **1**,335–341 (2004).
[12] A.G. Ruzzelli, G.M.P.OHare, R. Tynan, P. Cotan and P.J.M Havingat, Protocol Assessment Issues in Low Duty Cycle Sensor Networks: the Switching energy. Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy, **1**, 136–143 (2006).
[13] IEEE802.15.4–2006, Part 15.4: Wireless medium access control (MAC) and physical layer (phy) specification for low-rate wireless personal area networks (wpans), Sept. (2006).
[14] P. Levis, S. Madden, D. Gay, J. Polastre, R. Szewczyk, A. Woo, E. Brewer and D. Culler, The Emergence of Networking Abstractions and Techniques in TinyOS, Proceedings Of NSDI, **1**, 1–14 (2004).
[15] X. Chen, W.J Chen, P. Han, S.L Tu and Z.L Chen, Multi-channel Media Access Control Protocol for Wireless Sensor Networks, Journal of Chinese Computer Systems, **28**, 1729–1734 (2007).
[16] D. Gay, P. Levis, R.von Behren, M. Welsh, E. Brewer and D. Culler, The nesC language: A holistic approach to networked embedded system, Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation, **1**, 1–11 (2003).

Long Chen received B. S. degree from Jiangsu Institute of Technology, Jiangsu, China, in 1982, and the Ph.D. degree from Jiangsu University, Zhenjiang, China, in 2006. He is a professor in the School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang, China. His research interests include vehicle dynamic performance, traffic management and road safety, and logistics technology.

Xiao-jun Li received B. S. degree in Shanxi Normal University, Linfen, China. At present, he is a graduate student in the School of Computer Science and Communication Engineer, Jiangsu University, Zhenjiang, China. His research interests include cryptographic protocol and vehicular ad hoc networks.

Yuan-bo Guo received his B.S. and M.S. degree in Computer Science from the Information Engineering University, Zhengzhou, China, in 1997 and 2000, and the Ph.D. degree in Computer Applications and Technology from Xidian University, Xi'an, China, in 2005. Currently, he is an associate professor in the school of electronic technology, Information Engineering University, and his research interests include systems survivability and wireless network security.

Liang-min Wang received B. S. degree in computational mathematics in Jilin University, Changchun, China, in 1999, and the Ph.D. degree in Cryptology from Xidian University, Xi'an, China, in 2007. He is an associate professor in the School of Computer Science and Communication Engineer, Jiangsu University, Zhenjiang, China. Now his research interests include security protocols and wireless sensor networks. Dr. WANG is a senior member of Chinese Computer Federation.