Journal of Network and Computer Applications 58 (2015) 155–164

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

# DS-MAC: An energy efficient demand sleep MAC protocol with low latency for wireless sensor networks

Guijuan Wang [a], Jiguo Yu [a,*], Dongxiao Yu [b], Haitao Yu [c], Li Feng [d], Pan liu [a]

[a] School of Information Science and Engineering, Qufu Normal University, Rizhao, Shandong 276826, China
[b] Department of Computer Science, The University of Hong Kong, Pokfulam, Hong Kong, China
[c] College of Computer Science and Technology, Harbin Engineering University, NanGang District, Harbin, China
[d] Faculty of Information Technology, Macau University of Science and Technology, Macau, China

ARTICLE INFO

*Article history:*
Received 14 February 2015
Received in revised form
5 September 2015
Accepted 30 September 2015
Available online 9 October 2015

*Keywords:*
Wireless sensor network
Duty cycling
Medium access control
Delay
Energy efficiency

ABSTRACT

Duty cycling mechanism has been widely used to conserve energy that consumed by idle listening in wireless sensor networks, while fixed duty cycling introduces transmission latency in packet delivery. End to end latency is one of the most significant factors of packets loss in wireless sensor nodes, and many techniques have been proposed based on listening adaptively to reduce delay, which are mainly designed for light traffic loads. In this paper, we propose a novel asynchronous duty cycling MAC protocol, called demand sleep MAC (DS-MAC) that allows nodes to adjust their sleep time adaptively according to the amount of the received data packets in order to efficient and effective communication in the dynamic traffic load. DS-MAC protocol attempts to transmit a series of short token packets to wake up the receiver, which avoids the overhearing problem. Nodes in DS-MAC put the prediction field into ACK packets, which decreases the waiting delay of source node. Comprehensive simulation shows that when there are variable flows, such as broadcast traffic or transmissions from hidden nodes, DS-MAC significantly decreases waiting delay and energy consumption.

## 1. Introduction

Wireless sensor networks (WSNs) can be used to monitor the occurrence of many rare events, such as melting glacier, forest fire, and others. Wireless sensor nodes in environmental monitoring applications are often placed in hard-to-reach places of the valley and so on. Once the energy of a node exhausted, it is difficult to replace the battery, and the battery constrained energy limits the lifetime of the network. Idle listening is one of the most significant factors of energy consumption in wireless sensor nodes. Hence, many solutions for saving energy have been proposed utilizing the technique of duty cycling, just like B-MAC (Polastre et al., 2004) and S-MAC (Ye et al., 2002), when nodes have no data to transmit, it will turn their radio off to sleep for saving energy. Using duty cycling mechanism, each sensor node periodically switches between active state and sleep state. In the active state, a node is able to monitor the channel and transmits or receives data. In the sleep state, node turns off its radio to switch to a low power consumption mode. However, the wakeup–sleep mechanism brings about end-to-end transmission delay, especially when the traffic load is heavy. While the fixed duty cycling brings the end-to-end transmission latency, it also leads to the low bandwidth utilization ratio. As a part of the data link layer, the medium access control (MAC) layer controls the way that how the wireless sensor node to send or receive information, and the way to access the shared wireless medium. An efficient MAC protocol can reduce collisions, decrease end-to-end delay, increase network throughput and the lifetime of network. In order to reduce delay, save energy and improve the throughput, a variety of MAC protocols have been proposed.

Roughly speaking, MAC protocols with duty cycling technique for WSNs can be categorized into synchronized and asynchronous mechanisms, along with some other hybrid combinations. The purpose of these protocols is to reduce the idle listening and save energy. For rare event monitoring, idle listening is the main energy consumption state. Many synchronized protocols, such as S-MAC (Ye et al., 2002), T-MAC (Van Dam and Langendoen, 2003), R-MAC (Du et al., 2007), DW-MAC (Sun et al., 2008a), wake up nodes at the same time to communicate by synchronizing each sensor node. However, synchronous duty cycling MAC protocols require multi-hops time synchronization, which causes large network control overheads and poor network scalability. Especially when the change probability of traffic loads is high, the fixed duty

---

* Corresponding author.
*E-mail addresses:* guijuan_wang@126.com (G. Wang), jiguoyu@sina.com (J. Yu), dxyu@cs.hku.hk (D. Yu), ALbertyht@163.com (H. Yu), lfeng@must.edu.mo (L. Feng), liupan031189@163.com (P. liu).

http://dx.doi.org/10.1016/j.jnca.2015.09.007
1084-8045/

cycling approach is inefficient in the performance of latency and bandwidth utilization.

While, nodes in the asynchronous approaches such as B-MAC (Polastre et al., 2004), WiseMAC (Amre and Jean-Dominique, 2004; Dutta et al., 2010; Niu et al., 2013; Tang et al., 2013) can sleep or wake up on its own duty cycle schedule without the constraint of synchronization mechanism. Nodes in Polastre et al. (2004), prior to data transmission, transmit a preamble which length is at least as long as the length of the receiver's sleep period. If a node finds that the packet is not expected for itself, it will go back to sleep state to save energy. When a sender has data to transmit, the sender utilizes low power listening (LPL) to connect the receiver. It is not necessary for explicit synchronizing between a sender and a receiver, the receiver just needs to wake up for a short time to sample the channel, which decreases the idle listening time and saves energy.

However, asynchronous MAC protocols also produce many questions. Firstly, too long preamble causes the overhearing problem of the non-target node. Secondly, asynchronous mechanism brings end-to-end waiting delay, since the communication nodes must be awake. Huang et al. (2013) reviewed the improvement and development, in the terms of preamble and delay, of asynchronous MAC protocols. Thirdly, a fixed duty cycling mechanism is not suitable for the changeable traffic load. Since too long sleep time will bring the packet queuing delay and too short sleep time will increase the idle listening time, which resulting in a waste of energy. Fourthly, when queuing packets are too many, it will cause packets' overflow.

In this paper, we propose a novel asynchronous duty cycle MAC protocol, called demand sleep MAC protocol (DS-MAC). DS-MAC introduces a novel mechanism, namely, demand sleep method, which adjusts the node's sleep time adaptively according to the amount of received data packets. When the amount exceeding a threshold, DS-MAC will shorten node's sleep time, since frequently switching between sleep state and active state will waste much energy. When the energy that a node saved in the sleep state is less than a node consumed when changing between sleep state and active state, for saving energy, the node will not turn into sleep state any more in the duty. When the amount of data packets node received is less than the threshold, nodes will increase its sleep time to save energy that wasted by idle listening. Similar to CMAC (Liu et al., 2009), DS-MAC is an asynchronous MAC protocol, which uses a sequence of token packets as preamble to wake up the destination node and also uses token packet as ACK packet. Different from CMAC, DS-MAC devises a prediction mechanism to estimate a node's wake-up time by putting the node's sleeping time of this duty into ACK and sending to its neighbors, therefore neighbors can know the node's wake-up time in next duty. To a certain extent, DS-MAC has the same effect with synchronous MAC protocol. Knowing the receiver's wake up time, the sender can decrease sending time of preamble, which shortens waiting latency. Concretely, the contributions of the paper are as follows.

1. We propose a novel asynchronous duty cycling MAC protocol, called DS-MAC, which adjusts node's sleep time adaptively according to the amount of received data packets. Whether in high or light traffic loads, nodes can receive or can send data timely, which reduces the waiting delay.

2. The overflow probability of queuing packages is decreased, since the waiting latency reduced, ultimately cutting down the loss of data packages.

3. DS-MAC adds the predicted sleep time into ACK package, which has the same effect with synchronous MAC protocol. With the prediction mechanism, a node only needs to wake up slightly before the receiver when it has data to transmit. Therefore, DS-MAC improves the energy efficiency and increases the lifetime of the network greatly.

4. Most of MAC protocols use such wireless sensor network model, which assuming that there is only single source sending data packets to sink node when an event happens. While DS-MAC does not restrict the number of source nodes that sending data packets, which enlarges the scope of application of DS-MAC.

The rest of this paper is organized as follows. In Section 2, we discuss related works in duty cycling MAC protocols for sensor networks about shortening transmission latency and decreasing overhearing problem, even about how to maintain the power efficiency. In Section 3, we present the system model of the protocol. In Section 4, we give a detailed design and analysis of DS-MAC protocol. Section 5 describes the impact of loss packet on the performance of DS-MAC. Section 6 compares DS-MAC with S-MAC and SW-MAC by simulation. The simulating scenarios include both constant bit rate (CBR) and variable bit rate (VBR) traffic patterns. We provide conclusions and discuss potential research work in Section 7.

## 2. Related works

The quality of MAC protocol is directly related to the performance of wireless sensor networks, since a MAC protocol can control the way that wireless sensor nodes accessing to the medium. A number of previous approaches have been proposed to adjust the duty cycling for saving energy or shortening the end-to-end latency in duty cycle MAC protocols. However, none of them like DS-MAC protocol can adjust the node's duty cycling adaptively according to the received data packets in changeable traffic loads, which significantly improves the network performance. In this section, we review the synchronous MAC protocols, and then the asynchronous MAC protocols.

### 2.1. Synchronous MAC protocols

Firstly, we describe the mechanism of synchronous protocols (Huang et al., 2013). Nodes in synchronous MAC protocol listen to the channel for a certain time. If a node does not hear any schedule from neighbor nodes, it decides its next wake up time and broadcasts its schedule to others, which makes the node become a synchronizer. If a node receives a schedule table from a neighbor before choosing its own schedule, it will follow the received schedule, which makes it a follower. Usually, these nodes are away from the cluster head node few hops. Nodes that all in one cluster are synchronized by a synchronizer that one hop or few hops away from them. If a node receives a different schedule table from neighbors after creating its own schedule, it will adopt two schedules, which allows it to become a bridge node between two clusters. Therefore, the node will wake up at the time of its own cluster header wakes up and adjacent cluster header wakes up.

S-MAC (Ye et al., 2002) is one of the most classical periodic synchronous MAC protocols. In S-MAC, the time of node is divided into some frames and then each is further divided into three periods, namely, SYNC, DATA and SLEEP. The wake up time and the sleep time of nodes in S-MAC are both fixied, not random. At the beginning of SYNC period nodes wake up to synchronize clocks with neighbors to ensure that the node and its neighbors wake up concurrently. Then, in the DATA period all nodes remain active. If a node has data packets to send, it exchanges Request-to-Send (RTS) and Clear-to-Send (CTS) frame with its destination node to contend for the channel. When source nodes receive the acknowledgement frame, they transmit packets to the destination node. For saving energy, if a node is not included in the communication or has no data to send, it will turn off its radio to sleep. It is evident that in every cycle, one packet can only be transmitted through one hop. Although later adaptive listening (Ye et al., 2004) was introduced to the S-MAC protocol to overcome these

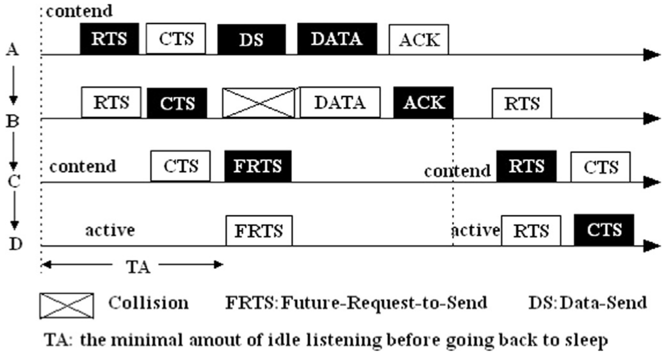

**Fig. 1.** The FRTS mechanism in T-MAC (Van Dam et al., 2003).

shortcomings, there is a certain increased restrictions. A data packet can be transmitted at most two hops in each cycle. In order to obtain the ambient energy resources, Tadayon et al. (2013) introduces a solar energy-harvesting model into S-MAC.

T-MAC (Van Dam and Langendoen, 2003) further improves adaptive performance of S-MAC protocol. In S-MAC, nodes will remain awake through the entire active period, even if they are neither sending nor receiving data. T-MAC improves S-MAC by listening to the channel for only a short time after the synchronization phase, and if no data is received during this window, the node returns to sleep mode. If receiving data, the node keeps awake until no further data is received or the active period ends. For saving energy, T-MAC uses one fifth of the energy, which is used by S-MAC. Meanwhile, T-MAC allows nodes to transmit two hops away. Fig. 1 shows the operation process of T-MAC with future Request-to-Send packet. When node **A** wants to send data to node **B**, it transmits the RTS packet first. Node **B** needs to reply an ACK packet after received the RTS packet. At the same time, neighbor node **C** overhears the CTS packet and then it sends a FRTS packet to its target receiver **D**. The FRTS packet contains the length of the current data transmission time from node **A** to node **B**, hence node **D** can learn its wake up time. When received the FRTS, **D** returns to sleep state. After the communication between **A** and **B**, node **D** wakes up to receive the packet may be sent by node **C**. When node **C** transmits FRTS packet, node **A** will postpone its data transmission to transmit a Data-Send (DS) packet with the same size of FRTS to node **B** to prevent any neighbor nodes from taking the channel. The DS packet does not include any useful information. Therefore, the collision of FRTS and DS at node **B** is not a problem. T-MAC thus can forward a packet to 3 hops per cycle. Although T-MAC extends the forwarding hops and shortens the latency, it increases energy consumption, since many nodes will keep awake other than the receiver. Inspired by the FRTS mechanism, RMAC (Du et al., 2007) presents a novel approach to reduce latency in multi-hop forwarding. Same as S-MAC, RMAC divides time into repeated cycles and each is further divided into three periods: SYNC, DATA and SLEEP. Instead of exchanging data during the DATA period, a control frame called pioneer frame (PION) is forwarded by multiple hops. The control frame informs nodes that on one routing path, about when to wake up in the SLEEP period. A PION not only serves as a RTS frame to request communication, but also confirms a request like a CTS frame.

DW-MAC (Sun et al., 2008a) is an important synchronous MAC protocol for high traffic loads. For instance, broadcast traffic or converge-cast traffic (Zhang et al., 2007) can increase the channel contention suddenly. DW-MAC combines the media access control and scheduling, which decreases the latency produced by wakeup–sleep mechanism. In order to ensure that data transmission is collision free at its intended receiver, DW-MAC introduces a new low-overhead scheduling algorithm that allows nodes to wake up on-demand during the sleep period of an operational cycle. Therefore, as traffic load increases, DW-MAC achieves low delivery latency.

A review and taxonomy of synchronous contention-based MAC protocol for delay-sensitive wireless sensor networks is given in (Doudou et al., 2014). It divides protocols into two main categories: static schedule and adaptive schedule. Adaptive schedule MAC protocols are further divided into four subclasses: adaptive grouped schedule, adaptive repeated schedule, staggered schedule, and reservation schedule. From this survey, we can know that many protocols can ensure low end-to-end delay, but none of them can provide delay guarantee for time-constrained applications.

### 2.2. *Asynchronous MAC protocols*

Node in asynchronous MAC protocol can choose its own active schedule dependently and do not need to pay the price for synchronizing neighbors. Asynchronous MAC protocols can achieve ultra-low duty cycle, but must find an efficient way to set up communication between the sender and the receiver. The asynchronous MAC protocol mainly uses preamble to wake the receiver up to transmit data, and the long preamble brings the overhearing problem. A non-target node only can find that it is not the destination node after received the long preamble completely. The overhearing problem brings energy waste of non-target nodes, which is proportional to the density of sensor nodes. The advantage of preamble sampling is that the channel sampling duration is short, therefore the channel sampling can be made frequent, which makes preamble sampling protocols having a ultra-low duty cycle. The performance of Aloha with preamble sampling has been analyzed in El-Hoiydi (2002).

B-MAC (Polastre et al., 2004) is the first typical asynchronous MAC protocol for wireless sensor network, which utilizes low power listening and an extended preamble to achieve low power communication. Nodes in B-MAC have an awake period and a sleep period, and each node has an independent schedule. If a node has data to send, it transmits a preamble first that is slightly longer than the sleep period of the receiver. During the active period, a node samples the channel and it sends an ACK packet to the source node if a preamble is detected. Furthermore, a node remains active until no data is received. In terms of throughput, latency and energy efficiency, B-MAC with the Low Power Listening (LPL) mechanism outperforms many existing synchronous MAC protocols.

Compared with B-MAC, a node in WiseMAC (Amre and Jean-Dominique, 2004) efficiently reduces the length of the preamble by learning the schedule of its direct neighbors. The receiver puts the time of its next awake time in the data acknowledgment frame, hence the sender can learn the schedule of the receiver active period, and schedules its transmission so as to reduce the length of the extended preamble. This reduces the overhearing and decreases the latency. Although WiseMAC solves many problems associated with low power communication, it does not provide a mechanism by which nodes can adapt to change traffic pattern. In order to avoid overhearing problem introduced by long preamble and reduce contention, delay, AS-MAC (Jang et al., 2013) asynchronously schedules the wakeup time of neighboring nodes.

To solve the problem brought by the long preamble, the asynchronous MAC protocol X-MAC (Buettner et al., 2006) creates a strobed preamble, which enables the target receiver to shorten the preamble by an early acknowledgement, thereby saving additional energy both at the sender and the receiver at the same time reducing per-hop latency.

CMAC (Liu et al., 2009) is a novel convergent MAC protocol, which achieves low latency by any-cast and convergent packet forwarding mechanisms. CMAC alleviates overhearing problem incurred by the long preamble mechanism of BMAC, it breaks up the long preamble into multiple RTS packets. When nodes wake up, they define the neighbor node which closest to the sink as the next hop. Thus, the density of nodes has a great influence on the performance of CMAC protocol. In addition, CMAC protocol is not suitable for the variable data traffic. Protocols that share the similar design including CSMA-MPS (Lin et al., 2004), DPS-MAC (Dual Preamble Sampling MAC) (Lim et al., 2006), TICER (Bernardo et al., 2007), MH-MAC (Wang et al., 2007). DPS-MAC (Lim et al., 2006) and MH-MAC (Wang et al., 2007) also include timing information for broadcasting message, allowing receivers to go back to sleep and wake up at the beginning of the data transmission.

Although nodes in WiseMAC can obtain more time for data transmission by learning the neighbor's wakeup schedule , the responsibility of establishing communication is gradually shifted from the sender side to the receiver side, since the overhead of learning receiver's schedule is more and more. Just like RI-MAC (Sun et al., 2008b), Koala (Liang et al., 2008), AS-MAC (Jang et al., 2013), and A-MAC (Dutta et al., 2010) improves throughput by receiver-initiated design. When the traffic load is light, nodes in RI-MAC will send a base beacon which only includes a source field without the back-off window (BW) size. After the receiver receipted the base beacon, it will send data packets immediately to the sender, which can optimize the low traffic load scenarios. When traffic load is heavy, node will broadcast a beacon that contains a BW field, each sender random back-offs according to the BW field. On detecting collisions, the receiver increases the value of the BW field until it reaches the maximum window size. RI-MAC can adjust node about how to send beacon in two kinds of traffic loads, however, collision still is the major factor that bounds throughput, since contention among neighbor nodes is synchronized by each beacon.

In order to quickly and steadily set up communication between a sender and a receiver, ORW (Ghadimi et al., 2014) takes advantage of the novel opportunistic routing metric EDC. Nodes in ORW obtain the routing table before transmitting data to neighbors. Firstly, node broadcasts a probe message to the neighbor nodes. Then, neighbors reply it with ACK information after received data. After receiving confirmation information, the sender adds the interface information into routing table and updates the table. Therefore, before sending data packets the sender obtains the forwarder sets. In addition, nodes use overhearing to estimate the quality of the link and ensure that only one pair of nodes communicating in the channel. ORW reduces delay and improves energy efficiency, while it focuses on the low data rate communication. Therefore, it is not well suited for the dynamic traffic loads.

Anchora et al. (2014) is also a duty cycling asynchronous MAC protocol. Nodes in the protocol schedule its own wake-up intervals and avoid useless awakening by exploiting the periodic transmission information of neighbor nodes. The protocol is robust, since every time when nodes detect a variation about neighbors, it will update its information table of neighbors. An analytical model is proposed in the protocol and it can predict the system behavior accurately. Finally, the MAC protocol can significantly reduce the power consumption in a WSN, while it does not fully take the performance of delay into account. SW-MAC (Liang et al., 2014) is a new asynchronous MAC protocol, nodes in SW-MAC use scout packet as preamble which plays the role of RTS and CTS. When a node has data to transmit, it sends a series of short scout packets. If the scout is received by a destination node, the receiver will reply the sender with an acknowledge packet, therefore the communication between a sender and a receiver is set up. Different from other asynchronous MAC protocols, SW-MAC is designed for the dynamic source node traffic rate. It alleviates the latency by adjusting the sleeping window size adaptively. However, SW-MAC is not reliable since nodes only according to the waiting time interval to determine the transmission rate. When the transmission interval turns large due to the loss of packets, it is not suitable to decrease the sleep window. In addition, SW-MAC does not consider the problem that multi-source nodes increase the traffic load. SW-MAC shrinks the scope of application, since it confines the number of source node as only one.

While, the number of source nodes is not limited in DS-MAC protocol, thereby increases the range of application. Although the DS-MAC also uses duty cycling to reduce the overhearing problem, the duty cycling of DS-MAC is not fixed, but dynamic. Asynchronous MAC protocols usually lead to the end-to-end delay by utilizing the preamble to wake the target node up, thus DS-MAC is designed to reduce latency. It sends a series of short token packets to wake the target node up, which solves the overhearing problem and reduces waiting delay by the prediction mechanism in dynamic traffic loads. When the network is congestion, nodes in DS-MAC reduce their sleep time according to the amount of received data packets, optimizing the network condition. In this case, DS-MAC protocol outperforms the bulk of asynchronous MAC protocols.

## 3. System model

In order to achieve energy efficiency, most of asynchronous MAC protocols have adopted the periodic wakeup–sleep mechanism to set up the communication between senders and receivers. While the wakeup–sleep method produces the end-to-end latency, since nodes are involved in the communication must be in active state, therefore a node that has data to send has to wait until the target receiver wakes up. For alleviating the latency, we adjust sleep time of nodes adaptively and add the predicted sleep time of a cycle into acknowledgement packets. By prediction mechanism, the sender can predict the wake-up time of target receiver, so that a sender only needs to wake up slightly earlier than the target receiver, which can decrease the waiting latency and keeps energy efficiency.

We adopt the following wireless sensor networks mode. Nodes are randomly and densely distributed in the detected area. Each node is response for both detecting events and forwarding data as a router. When an event occurs, the node encapsulates and transmits the data by multi-hop to the destination node or sink node. We assume that the wireless sensor network is connected. In the routing layer, there are one or more paths between senders and receivers, so that each node has at least one multi-hop routing path to the destination node when an event occurs, which reduces the establishment latency. Therefore, considering the characteristics of MAC layer, to design an efficient routing protocol remains a challenging problem.

Many MAC protocols, such as SW-MAC (Liang et al., 2014), assume that there is only one source node with data forwarding to a sink node when an event occurs, which greatly reduces the available scope of this protocol. While, in this paper, we do not limit the number of source nodes that have data to send, but only need to adjust the sleep time of nodes adaptively according to the amount of received data. To save energy, we design DS-MAC as an asynchronous MAC protocol which is based on the duty cycle mechanism. When a node wakes up, it samples the channel firstly by using the outlier detection method of B-MAC, to decide whether the channel is idle or not. In DS-MAC, we only consider the simple detection method, which means that the event can be a single property or can be a single node detected. Other more complex detection methods can be found in Viswanathan and Varshney (1997), Yu et al. (2006) and Tan et al. (2010).

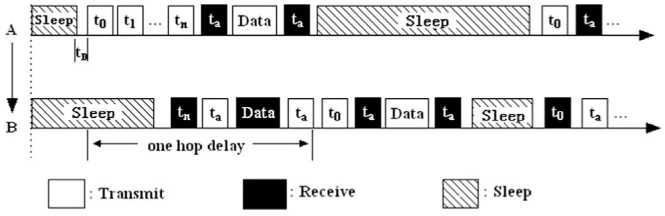

**Fig. 2.** DS-MAC system model.

Similar to SW-MAC (Liang et al., 2014), we divide the long preamble into a series of short token packets, which contains the target address of the next hop and the sequence number of the token packet, as shown in Fig. 2. Node **A** periodically wakes up to detect the duration of an event. When node **A** needs to send data, it continuously sends a series of token packets to wake up the receiver until it receives the ACK packet from destination node **B**. A gap between two short token packets is deliberately inserted, which allows the destination node **B** to reply with an ACK. Node **B** periodically wakes up to check if there are any arriving data frames intended for itself. When receiving the token packet, node **B** decodes the destination address information. If node **B** is the destination, it stays awake to receive the arriving data, otherwise it goes back to the sleep state for saving energy. After receiving data packets, node **B** needs to adjust sleep time dynamically according to the amount of received data. Different from other MAC protocols, nodes in DS-MAC put their sleep time of this duty into ACK packets, which achieves the effect that there is an appointment about next time communication. In the next section we will describe the design of DS-MAC protocol in detail.

## 4. Protocol design

### 4.1. Basic operation

In many practical applications, some performances such as, end-to-end latency, energy efficiency and traffic throughput, asynchronous MAC protocols outperform synchronous MAC protocols. An asynchronous MAC protocol does not require nodes that communicate with each other to share the scheduling information, therefore they can save a large amount of overhead. Asynchronous MAC protocols have the better scalability than synchronous MAC protocols. Hence, we adopt the asynchronous mechanism to design the DS-MAC.

However, an obvious problem in asynchronous MAC protocols should be considered, that is, when a node wakes up with data to send, it must wait until the receiver wakes up, which produces the waiting delay. When the traffic load becomes heavy, the packet will overflow, which causes severe packet loss problem. Nodes in DS-MAC according to the received data to adjust sleep time adaptively and put the wake-up time of next duty into the acknowledgement packet, which decreases the latency and saves the energy that consumed by sending token packets.

As shown in Fig. 2, nodes periodically convert between active state and sleep state. Here we describe the operations of nodes in the two states in detail.

*A*ctive state: When node **A** wakes up, it samples the channel during the time $t_D$ first, to decide whether there is data to be transmitted. When an event occurs, node **A** encapsulates the data and transmits it by multi-hops to the destination node **B** or sink node. Then node **A** sends a series of token packets to wake up the receiver **B**. A gap $t_a$ between two short token packets is deliberately inserted, which allows the destination node to reply with an ACK after received a token packet. Each token packet has its serial number. In the communication, the worst case is that the data packet has a maximum waiting delay, that is, the receiver **B** wakes up after the last token packet is forwarded. The total waiting time of source node **A** before sending the data packet is

$$T_{A,B}^{wait} \le \sum_{i=0}^{n}(t_s^i + t_a^i) = (n+1)(t_s + t_a) \tag{1}$$

where $t_s^i$ and $t_a^i$ are sending time of *i*th token packet and transmitting time of the *i*th token ACK packet, respectively. $t_s$ and $t_a$ are sending time of one token packet and transmitting time of one token ACK packet, respectively. Since all token packets spend the same transmission time and all ACK packets also spend the same transmission time, after the token packet is received successfully, the waiting time of data packets is equivalent to the $n+1$ times of the time that transmitting one token packet and receipting one acknowledgment packet.

Therefore, the time of sending one data packet is

$$T_{data} = t_d + t_a \tag{2}$$

where $t_d$ is the time of sending one data packet, generally it is the largest time in all of the transmission and $t_a$ is the time of transmitting one ACK packet.

From Eqs. (1) and (2), the total time of sending one data packet is $T_{one}$ :

$$T_{one} = T_{A,B}^{wait} + T_{data} \tag{3}$$

where $T_{one}$ also denotes the one hop delay.

When a node wakes up finding that there is no event, then node goes back to sleep state. In this case, the listening time of the node is equal to the active time.

$$T_{listen} = t_D + t_s^l + t_e \tag{4}$$

where $t_D$ denotes the time of sampling channel to decide whether there is an event or not. $t_s^l$ is the time of receiving and decoding an token packet that is not sent for it. $t_e$ is the extra waiting time to prevent the node immediately goes back to sleep state after receiving the token packet that is not intended for it. During this period, the node needs to decide if there are other nodes contend for the channel to send data to it.

*Sleep time*: Nodes in asynchronous MAC protocols usually select their sleep time according to the random selection mechanism in interval $[T_{sleep}^{min}, T_{sleep}^{max}]$. Where $T_{sleep}^{min}$ and $T_{sleep}^{max}$ denote the minimum sleep time and maximum sleep time, respectively. While, the sleep time of nodes in DS-MAC is not fixed and has not scope, it is adjusted adaptively by the amount of received data. In order to save energy, the sleep time of the node is $T_{sleep}^{max}$ when it wakes up for the first time. In this paper, it is not necessary to use ultra-low duty cycling to save energy, since the frequent switching of a node between active state and sleep state will consume much energy. When the saved energy during sleep state is less than the consumed energy by frequently switching between active state and sleep state, the node will not go to sleep.

### 4.2. Energy consumption of DS-MAC

The energy consumption of nodes in wireless sensor networks in each state is shown in Fig. 3 (Shnayder et al., 2004). We use $e_s, e_w, e_l$ and $e_t$ denote the energy consumption in sleep state, wake-up state, listening state and transmitting state, respectively. We can know the relationship of energy consumption in every state is $e_s < e_w < e_l < e_t$. When the energy consumption of switching between sleep state and active state is less than the saved energy by sleeping, the node will go to sleep. That is

$$T_{sleep}e_s + T_w e_w \le T_{sleep}e_l$$

$$T_{sleep} \ge \frac{T_w e_w}{e_l - e_s} \tag{5}$$

where $T_{sleep}$ and $T_w$ denote the minimum sleep time of a node and

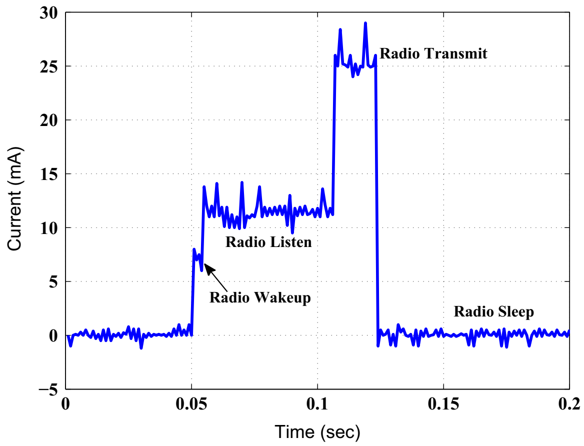

**Fig. 3.** Energy consumption in Mica2 (Shnayder et al., 2004).

the time that a node switches from sleep state to the active state, respectively. In this paper, we set the $T_w$ as a fixed value.

### 4.3. Setting sleep time dynamically

We assume that there is a counter in each node to calculate the amount of received data, let $n$ be the number of received data packets. On receiving one packet, the counter will automatically add 1. Before going back to the sleep state, the node computes the time consumed by receiving data packets. From Eq. (3)

$$T_{sum} = T_{one} \times n \tag{6}$$

We set the total length of a duty as $T$ and the rest time of one duty cycle as $T_{rest}$,

$$T_{rest} = T - T_{sum} - T_c \tag{7}$$

where $T_c$ is the time consumed by calculating the remaining time of the duty cycle, by which the node judges to decide whether to sleep or not. Usually, we set $T_c$ as a constant according to the computing ability of nodes. When the node's computing ability is high, we set the $T_c$ a small value. When the node's computing ability is low, we set $T_c$ a relatively large value.

Now we describe about how the node to choose their sleep time in detail. Although nodes in DS-MAC select their sleep time adaptively, there we give the maximum sleep time $T_{sleep}^{max}$. That is , when a node samples the channel, and finds there is no data to forward, then the node goes back to sleep state.

$$\begin{aligned} T_{sleep}^{max} &= T - (t_D + t_s^l + t_e) \\ &= T - T_{listen} \end{aligned} \tag{8}$$

(1) When $T_{rest} \le T_{sleep}$, it indicates that the remaining time is less than or equal to the minimum sleep time, which shows that the traffic load is heavy and the energy saved by sleeping is not more than the energy consumed by switching between active state and sleep state. If the node goes back to sleep state, it cannot achieve the purpose of saving energy. Therefore we set the value of sleep time as $T_s = 0$, and then the node will remain active.

(2) When $T_{sleep} < T_{rest} < T_{sleep}^{max}$, which indicates that after receiving data packets the remaining time is greater than the minimum sleep time but less than the maximum sleep time. In this case the network traffic load is relatively high, so we set the sleep time as $T_s = T_{rest}$.

(3) When $T_{rest} > T_{sleep}^{max}$ which indicates that after the node wakes up to sample the channel, it goes back to sleep immediately when it finds out that it is not the target receiver of current transmission. Then we set the value of sleep time as $T_s = T_{sleep}^{max}$.

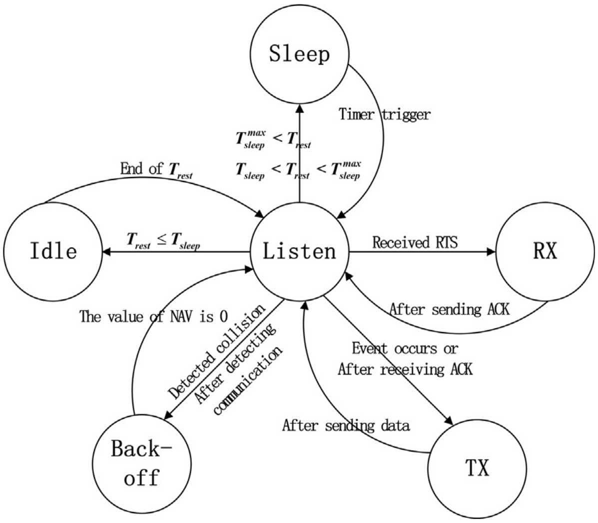

**Fig. 4.** Node's state in DS-MAC.

| 2bits | 14bits | 16bits | 16bits | 16bits | 16bits | 16bits |
|---|---|---|---|---|---|---|
| Type | Sequence | Src | Dest | Duration | Prediction | Variance |

**Fig. 5.** ACK packet format.

Figure 4 shows the transmission state of nodes in DS-MAC protocol. Where idle denotes a special state, in which when the remaining time of a node does not satisfy the condition of sleep, it will be in the idle state. When nodes detect one communication in channel or there is an collision in channel, then it will back off.

### 4.4. Node's operation in DS-MAC

In asynchronous MAC protocols, nodes choose their wakeup–sleep duty cycle independently. They do not need to exchange synchronous information with their neighbors. Therefore the possibility of nodes wake up at the same time is small. However, in a dense wireless sensor network, it is possible that when an event occurs there are many nodes wake up to sample data and forward data packets. To solve this problem, DS-MAC puts the duration of transmission between two communication nodes into token packets and put the prediction time of sleeping into ACK packet. The packet format is shown in Fig. 5.

Duration field shows the duration of transmission between a sender and a receiver. When a node founds that there are communication of neighbors in the channel, it will execute the back-off algorithm of CSMA/CA and set the value of network allocation vector (NAV). NAV is a mechanism that predicts the future communication of the channel. When the value is unequal to 0, which indicates that the channel is busy and the node goes to sleep immediately. During sleep, the NAV timer keeps working until the value of NAV is reduced to 0. When the value of NAV is 0 and the node is in the active state, it will compete the channel for forwarding data packets, while if the node is in sleep state, it will keep in the sleep state and does not need to wake up.

Figure 6 shows the operation of nodes in DS-MAC. Node **B** communicates with node **C**, node **A** that is adjacent to node **B** overhears the token packet coming from node **B** and decodes the duration information, then **A** sets the value of NAV as the time equal to the communication time between **B** and **C**. When the value of NAV is 0, the node **A** still be in the sleep state, then it remains sleeping. When node **D** listening to the acknowledgment packet of neighbor node **C**, it decodes the duration information and sets the NAV value. When the value of NAV decreased to 0, node **D** still be in the active state, then it will receive the data coming from node **C**.

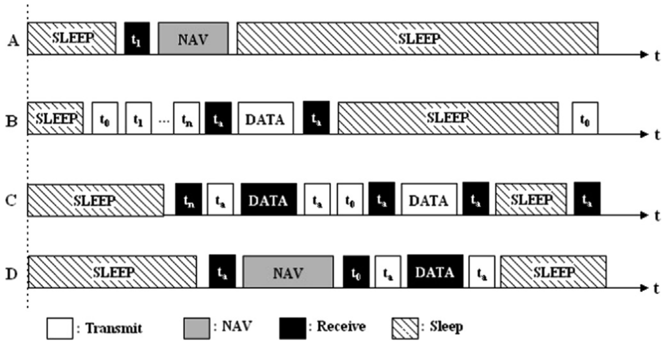

**Fig. 6.** Token-based scheduling.

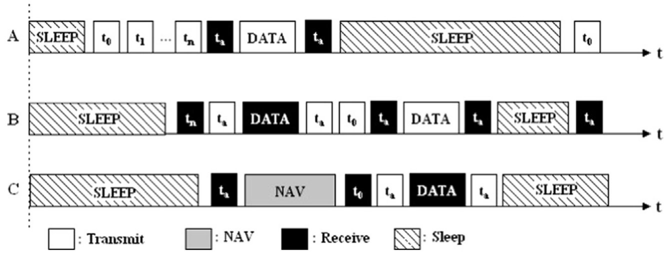

**Fig. 7.** The process of node **A** communicates with node **B** according to the prediction field.

The prediction field in Fig. 5 describes the prediction time of sleep in this duty. It is obtained by calculating the recorded twice sleep time. Since the sleep time is dynamic, node cannot obtain it accurately. According to the proportion of twice recorded sleep time, we calculate the prediction time. Then we put the prediction time into ACK packet to inform the source node. When a node has data to send in next cycle, the source node can refer to this value, in order to reduce unnecessary waiting time. Define the predict time as

$$T_{predict} = \alpha T'_s + \beta T''_s \tag{9}$$

where $T'_s$ and $T''_s$ are the sleep time of last time and the sleep time before last time of a node, respectively. Both $\alpha$ and $\beta$ are the weighting factors. $\alpha$ denotes the proportion of last time sleep time. $\beta$ denotes the proportion of sleep time before last time. Since the sleep time of node is dynamic, the reference value of $T'_s$ is relatively large, therefore we let $\alpha = 0.7$. The reference value of $T''_s$ is relatively small, therefore we set its value as 0.3. Since the prediction field is obtained by computing twice of sleep time, we set the value of prediction in the first two cycles as null. Nodes wake up and sleep according to the mechanism that without the prediction method. Nodes in DS-MAC from the third cycle to send data, wake up and sleep according to the prediction method.

By this prediction field, node's communication process is shown in Fig. 7.

When node **A** decodes the prediction field included in ACK packet, it knows the wake-up time of node **B** in next cycle. Node **A** turns into sleep state at the end of this communication. While node **C** wakes up after the communication between **A** and **B** to receive the data coming from node **B**, since the NAV is reduced to 0. At the end of communication between node **B** and node **C**, **B** goes back to sleep state according to the decision. In next duty cycle, node **A** only needs to wake up slightly before the node **B** to communicate with **B** according to the prediction field, which decreases the waiting delay, saves the energy consumed by sending token packets and avoids the problem that a sender and a receiver occupy the wireless medium too long time which hinders the communication among other nodes.

### 4.5. *Prediction error revise*

Since DS-MAC is mainly used in dynamic networks with changeable traffic loads, therefore the duty cycling is not a fixed value in the protocol. There is often error in the prediction of node's sleep time. When a node finds that the prediction error (defined as the difference between the estimated sleep time of the receiver and the actual sleep time of the receiver) is greater than a given threshold, which indicates that the prediction method can not decrease the waiting latency, the node is required to restart the prediction mechanism. Here we set the maximum sleep time $T^{max}_{sleep}$ as the threshold. When restarting the prediction mechanism, DS-MAC sets the value of prediction as null in the first two periods.

Nodes in DS-MAC will communicate with neighbors following the procedure.

Table 1
The proposed DS-MAC protocol

```
/* When node A has data for forwarding to node B; */
1. Node A detects the value of prediction;
2. if prediction < = threshold then
3.   Node A uses the prediction field to predict the wake up time of node B;
4.   if Node A listens and finds the channel is idle then
5.     Node A sends token packets to node B;
6.     if Node B receives the token packet then
7.     Node B replies node A with a token;
8.     Node A sends data packets to node B;
9.     end if
10.  else if There is communication in channel then
11.      Node A sets the value of NAV according to the communication;
12.      if The value of NAV is 0 and node A is in active state, then
13.        Node A sends data to node B;
14.      else
15.        Node A keeps sleep state;
16.      end if
17.   end if
18.   end if
19. else
20.   Node A sets the value of prediction as 0;
21.   Node A and node B execute step 4 to 18;
22. end if
/* After node B receives data packets; */
23. if $T_{rest} < = T_{sleep}$, then
24.   Node B keeps active state;
25. else
26.   Node B goes to sleep;
27. end if
```

## 5. Packet loss

Detailed analysis can be found in SW-MAC about the influence of missing packets due to the instability of wireless sensor networks. There are three kinds of lost packets, namely, *scout*, *acknowledgment* and *data*. Each lost packet needs to be retransmitted, which leads to the transmission delay and energy consumption. Although the problem of packets lost both present in DS-MAC protocol and SW-MAC protocol, while the impact of this problem on the two protocols is very different.

Nodes in SW-MAC protocol adjust the sleep time according to the arriving interval of packets. When receiving a data packet needs to wait for a long time, SW-MAC considers that the traffic loads is heavy and reduces the node sleep time, which consumes more time and more energy to receive data packets. In this case, the competition about accessing channel is more intense, which leads to more collisions and makes the network situation worse. When the wireless sensor network is in poor condition, resulting in the loss of scout packet, scout acknowledgment packet and data packet, which decreases the traffic loads. At the moment, nodes decrease their sleep time, which wastes more energy.

While, nodes in DS-MAC protocol adjust sleep time adaptively according to the amount of received data packets. The receiver increases its sleep time, since it has not received data packets for a long time, which saves the waiting time. Due to the packet lost, a receiver has not receive a packet for a long time, which indicates that the network condition is not suitable for communication, thus the receiver increases its sleep time to save energy and avoids the congestion of channel. SW-MAC protocol handles the packet loss by decreasing sleep time to retransmit data packets, which consumes more energy and may make the network condition becoming worse or even leading to the network paralysis.

DS-MAC protocol outperforms SW-MAC protocol, on the performance of improving the network status under the condition that packet loss especially in heavy traffic loads.

## 6. Simulations

In this section, we compare DS-MAC protocol with SW-MAC protocol and S-MAC protocol by NS2 (version 2.29) simulator under constant bit rate (CBR) traffic load and variable bit rate (VBR) traffic load about the performance of end-to-end latency and energy efficiency. The standard combined free space and two-ray ground reflection radio propagation model are used in simulation. Each sensor node is equipped with a single Omni-directional antenna.

Table 2 summarizes the MAC protocol parameters used in our simulations. In order to compared with SW-MAC protocol, we set 200 nodes randomly distributed in 2000 $m^2$ area and the communication range is set to 200 m. The source nodes transmit data packets to sink nodes.

SW-MAC is a new asynchronous MAC protocol and outperforms S-MAC protocol and CMAC protocol in many aspects. We compare DS-MAC with SW-MAC and S-MAC on the performance of end- to-end latency and energy efficiency. In SW-MAC protocol, there is no maximum number of retransmission times, there we set to 5. A receiver in DS-MAC adjusts the BW value in each beacon using a binary exponential back off with values of 0–255.

**Table 2**
Simulation parameters.

| Parameter | Value |
|---|---|
| *Simulation time* | 200 s |
| *Location of source* | (0,0) |
| *Bandwidth* | 38,400 bps |
| *Initial energy* | 1000 J |
| *BW* | 0–255 |
| $e_s$ | 0.003 w |
| $e_w$ | 0.020 w |
| *Monitor area* | 2000 $m^2$ |
| *Location of sink* | (1900, 1900) |
| *Length of data* | 100 bytes |
| $T_{listen}$ | 0.03 s |
| $e_t$ | 0.090 w |
| $e_l$ | 0.075 w |
| $T_w$ | 0.005 s |

### 6.1. Simulation results with CBR traffic

(1) *The performance of end-to-end latency*: SW-MAC considers packet arrival interval time as the main basis for adjusting the node's sleep window. From Figs. 8 and 9 we can know that when the network traffic load is light, SW-MAC protocol has a little delay

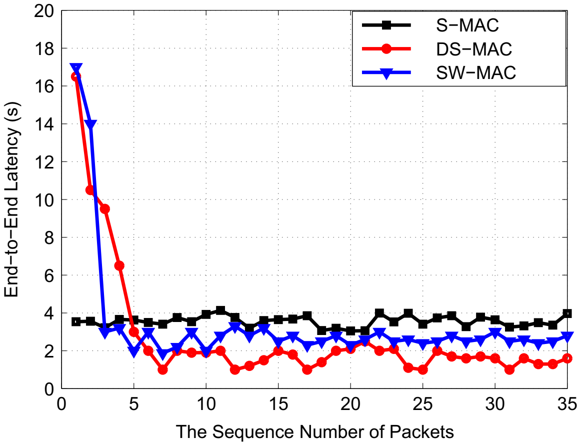

**Fig. 8.** End-to-End delay of light traffic loads.

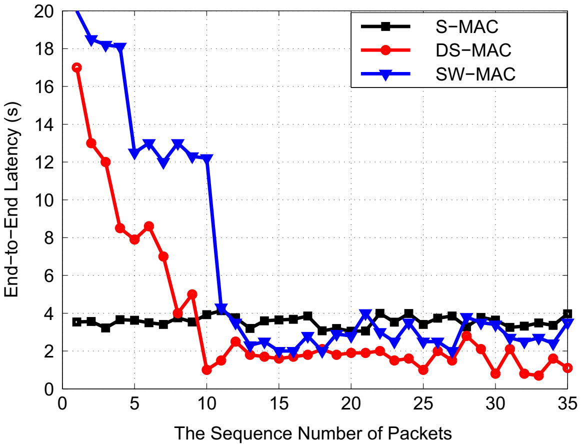

**Fig. 9.** End-to-End delay of heavy traffic loads.

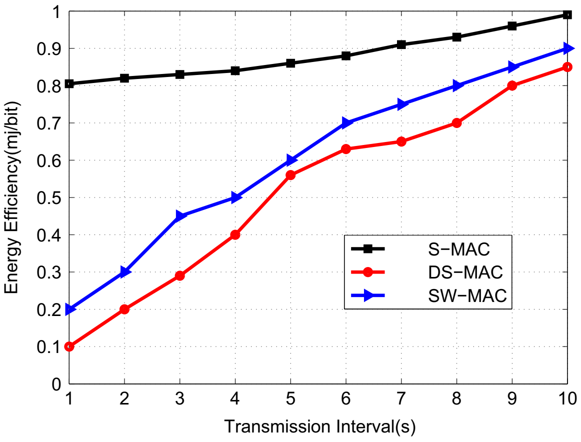

**Fig. 10.** Energy consumption of nodes.

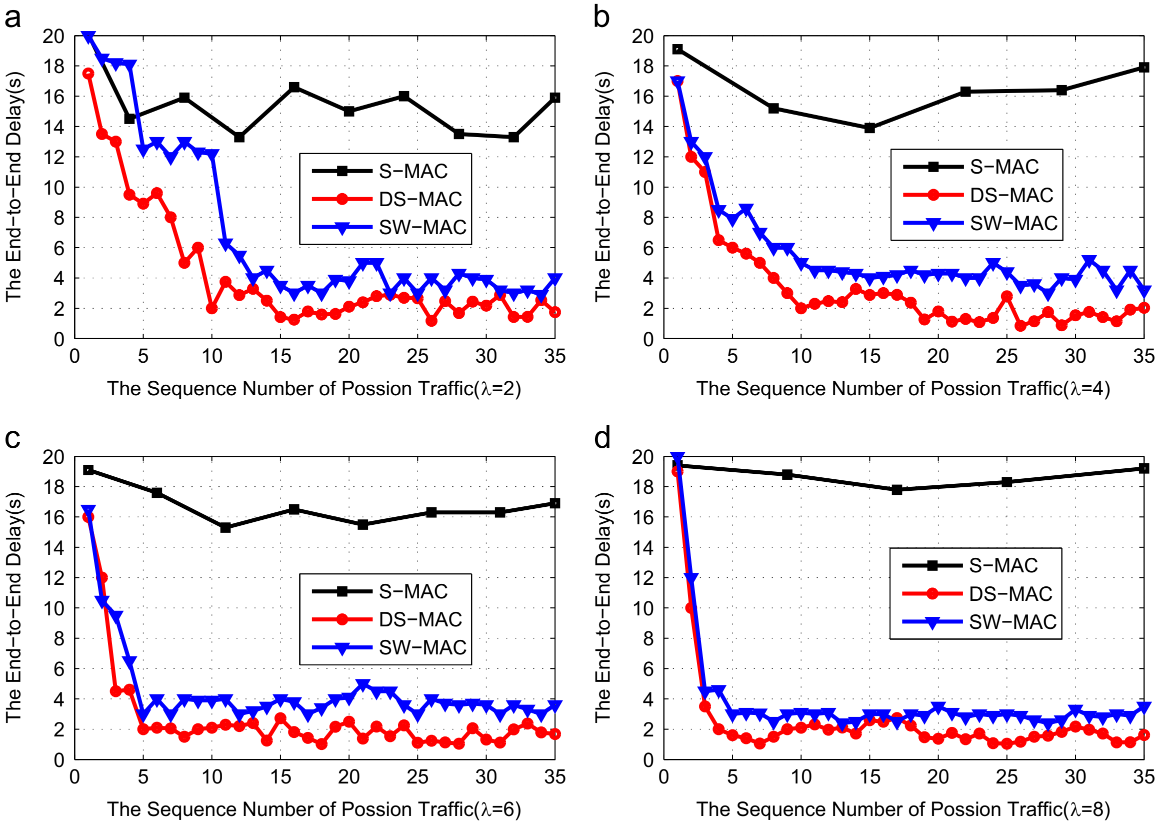

**Fig. 11.** The end-to-end delay under Possion traffic.

in transmitting data packets. When the network traffic load is heavy, SW-MAC decreases the node sleep time and retransmits lost data packets, which leads to the network condition being worse and results a higher latency. However, DS-MAC adjusts node's sleep time adaptively, which outperforms SW-MAC on the performance of transmission latency.

As shown in Fig. 8, since S-MAC is a synchronous MAC protocol, nodes in S-MAC know their wakeup–sleep cycle of neighbors, and the transmission latency is less than asynchronous MAC protocols at the beginning of communication. Both SW-MAC and DS-MAC are asynchronous protocols, at the beginning, the transmission latency is relatively large. While using the prediction mechanism after a period of transmission, the delay of DS-MAC is significantly reduced. Therefore the communication latency of DS-MAC is less than the S-MAC protocol in the whole.

We can know from Fig. 8, when the traffic load is light, although there is deference in the performance between SW-MAC and DS-MAC, the difference is little. While in Fig. 9, when the traffic load is heavy, SW-MAC protocol has relatively high fluctuations of end-to-end latency. In this case, the waiting packets are too many and resulting in the loss of packages. Therefore the node will increase its active time to receive the lost data, which further increases the burden of network and results in the deterioration of network condition. Therefore, SW-MAC needs a little long time to improve the quality of the network, for example, when the number of packets is 5–10, there has a sustained high latency. While, DS-MAC protocol can quickly improve network condition by adjusting sleep time adaptively, thereby reduces end-to-end latency. When the data rate is relatively stable and only increase traffic loads, there has little effect on the delay.

(2) *Energy efficiency*: As shown in Fig. 10, the consumed energy of S-MAC increases linearly when the communication time becomes longer and longer. When transmission rate of nodes decreases, the interval between two packets is more and more large, then SW-MAC protocol will decrease the sleep time, which consumes more and more energy. Meanwhile, SW-MAC frequently adjusts sleeping window consumed more energy in the changeful communication. However, nodes in DS-MAC protocol increase sleep time adaptively according to the amount of received data packets, which saves energy. DS-MAC protocol exceeds SW-MAC protocol in terms of energy efficiency.

### 6.2. Simulation results with VBR traffic

In order to evaluate the performance of DS-MAC protocol comprehensive, the following simulation is executed in VBR traffic mode. We assume the traffic follows a Poisson process with parameter $\lambda$. Since the main features of this paper is to reduce the transmission delay, which improves the network traffic condition. Here we analyse the performance of delay in detail. Figure 11 shows the performance of latency about DS-MAC protocol, SW-MAC protocol and S-MAC protocol under the different value of parameter $\lambda$. The value of $\lambda$ is 2, 4, 6 and 8, respectively. From the figure we can know that the delay in synchronous MAC protocol gradually increasing and in asynchronous MAC protocol gradually decreasing when the $\lambda$ increased.

End-to-end latency in the beginning of the communication is large in DS-MAC protocol, and then with the communication time increasing, the delay reduced. Since nodes in DS-MAC protocol can adjust sleep time adaptively according to the prediction mechanism, therefore reduces the transmission delay. The delay in DS-MAC protocol is lower than SW-MAC protocol.

## 7. Conclusion and further work

In this paper, we propose a novel asynchronous duty cycling MAC protocol DS-MAC, an energy efficient demand sleep MAC protocol with low latency for wireless sensor networks. The protocol is primarily designed for reducing the end-to-end latency in dynamic traffic loads. DS-MAC adjusts the sleep time of nodes adaptively according to the amount of received data packets, which reduces the end-to-end latency and saves energy. We put the Prediction field and Duration field into the ACK packet, which makes nodes back off to sleep state when it listens there has been communication in channel. In addition, nodes can predict the next wake-up time of their neighbors, which reduces the waiting time and further improves the performance of the network. However, there is deviation in predicting mechanism of our paper, especially in explosive communication. This is also a problem that will be considered in the future.

## Acknowledgements

This work is partially supported by the NNSF of China for contract 61373027, The Macau Science and Technology Development Fund under Grant no. 013/2014/A1, and NSF of Shandong Province for contract ZR2012FM023.

## References

Amre E, Jean-Dominique D. WiseMAC: an ultra low power MAC protocol for multihop wireless sensor networks. In: Proceedings of ALGOSENSORS; 2004. p. 18–31.

Anchora L, Capone A, Mighali V, et al. A novel MAC scheduler to minimize the energy consumption in a wireless sensor network. Ad Hoc Netw 2014;16:88–104.

Bernardo L, Oliveira R, Pereira M, et al. A wireless sensor MAC protocol for bursty data traffic. In: Proceedings of IEEE PIMRC; 2007. p. 1–5.

Buettner M, Yee GV, Anderson E, et al. X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In: Proceedings of ACM SenSys; 2006. p. 307–20.

Doudou M, Djenouri D, Badache N, et al. Synchronous contention-based MAC protocols for delay-sensitive wireless sensor networks: a review and taxonomy. J Netw Comput Appl 2014;38:172–84.

Du S, Saha AK, Johnson DB. RMAC: a routing-enhanced duty-cycle MAC protocol for wireless sensor networks. In: Proceedings of IEEE INFOCOM ; 2007. p. 1478–86.

Dutta P, Dawson-Haggerty S, Chen Y, et al. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. In: Proceedings of ACM SenSys; 2010. p. 1–14.

El-Hoiydi A. Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks. In: Proceedings of IEEE ICC 5; 2002. p. 3418–23.

Ghadimi E, Landsiedel O, Soldati P, et al. Opportunistic routing in low duty-cycled wireless sensor networks. ACM Trans Sensor Netw 2014;10(4):1–39.

Huang P, Xiao L, Soltani S, et al. The evolution of MAC protocols in wireless sensor networks: a survey. IEEE Commun Surv Tutor 2013;15(1):101–20.

Jang B, Lim JB, Sichitiu ML. An asynchronous scheduled MAC protocol for wireless sensor networks. Comput Netw 2013;57(1):85–98.

Liang L, Liu X, Wang Y, et al. SW-MAC: a low-latency MAC protocol with adaptive sleeping for wireless sensor networks. Wirel Pers Commun 2014:1191–211.

Liang CJM, Terzis A. Koala:2008—ultra-low power data retrieval in wireless sensor networks. In: Proceedings of IEEE IPSN; 2008. p. 421–32.

Lin EYA, Rabaey JM, Wolisz A. Power-efficient rendezvous schemes for dense wireless sensor networks. In: Proceedings of IEEE ICC 7; 2004. p. 3769–6.

Liu S, Fan KW, Sinha P. CMAC: an energy-efficient MAC layer protocol using convergent packet forwarding for wireless sensor networks. ACM Trans Sensor Netw 2009;5(4):1–34.

Niu J, Cheng L, Gu Y, et al. Minimum-delay and energy-efficient flooding tree in asynchronous low-duty-cycle wireless sensor networks. In: Proceedings of IEEE WCNC; 2013. p. 1261–6.

Polastre J, Hill J, Culler D. Versatile low power media access for wireless sensor networks. In: Proceedings of ACM SenSys; 2004. p. 95–107.

Shnayder V, Hempstead M, Chen B, et al. Simulating the power consumption of large-scale sensor network applications. In: Proceedings of ACM SenSys; 2004. p. 188–200.

Sun Y, Du S, Gurewitz O et al. DW-MAC: a low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks. In: Proceedings of ACM MobiHoc; 2008a. p. 53–62.

Sun Y, Gurewitz O, Johnson DB. RI-MAC: 2008—a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In: Proceedings of ACM SenSys; 2008b. p. 1–14.

Tan HX, Chan MC, Xiao W, et al. Information quality aware routing in event-driven sensor networks. In: Proceedings of IEEE INFOCOM; 2010. p. 1–9.

Tang H, Sun C, Liu Y, et al. Low-latency asynchronous duty-cycle mac protocol for burst traffic in wireless sensor networks. In: Proceedings of IEEE IWCMC; 2013. p. 412–17.

Tadayon N, Khoshroo S, Askari E, et al. Power management in SMAC-based energy-harvesting wireless sensor networks using queuing analysis. J Netw Comput Appl 2013;36(3):1008–17.

Van Dam T, Langendoen K. An adaptive energy-efficient MAC protocol for wireless sensor networks. In: Proceedings of ACM SenSys; 2003. p. 171–80.

Viswanathan R, Varshney PK. Distributed detection with multiple sensors I. Fundamentals. In: Proc IEEE 1997;85(1): 54–63.

Wang H, Zhang X, Nait-Abdesselam F, et al. DPS-MAC: an asynchronous MAC protocol for wireless sensor networks. In: Proceedings of HiPC; 2007. p. 393–404.

Ye W, Heidemann J, Estrin D. An energy-efficient MAC protocol for wireless sensor networks. In: Proceedings of IEEE INFOCOM 3; 2002. p. 1567–76.

Yu L, Yuan L, Qu G, et al. Energy-driven detection scheme with guaranteed accuracy. In: Proceedings of ACM IPSN; 2006. p. 284–91.

Ye W, Heidemann J, Estrin D. Medium access control with coordinated adaptive sleeping for wireless sensor networks. IEEE/ACM Trans Netw 2004;12(3):493–506.

Zhang H, Arora A, Choi Y, et al. Reliable bursty convergecast in wireless sensor networks. Comput Commun 2007;30(13):2560–76.