

Article

## An Energy-Efficient MAC Protocol Using Dynamic Queue Management for Delay-Tolerant Mobile Sensor Networks

Jie Li <sup>1</sup>, Qiyue Li <sup>2,\*</sup>, Yugui Qu <sup>1</sup> and Baohua Zhao <sup>2</sup>

<sup>1</sup> Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, Anhui 230027, China; E-Mails: myl@mail.ustc.edu.cn (J.L.); cnygqu@ustc.edu.cn (Y.Q.)

<sup>2</sup> Department of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China; E-Mail: bhzhao@ustc.edu.cn (B.Z.)

\* Author to whom correspondence should be addressed; E-Mail: liqiyue@mail.ustc.edu.cn; Tel.: +86-551-3607654; Fax: +86-551-3607654.

Received: 15 November 2010; in revised form: 9 December 2010 / Accepted: 5 January 2011 /

Published: 1 February 2011

---

**Abstract:** Conventional MAC protocols for wireless sensor network perform poorly when faced with a delay-tolerant mobile network environment. Characterized by a highly dynamic and sparse topology, poor network connectivity as well as data delay-tolerance, delay-tolerant mobile sensor networks exacerbate the severe power constraints and memory limitations of nodes. This paper proposes an energy-efficient MAC protocol using dynamic queue management (EQ-MAC) for power saving and data queue management. Via data transfers initiated by the target sink and the use of a dynamic queue management strategy based on priority, EQ-MAC effectively avoids untargeted transfers, increases the chance of successful data transmission, and makes useful data reach the target terminal in a timely manner. Experimental results show that EQ-MAC has high energy efficiency in comparison with a conventional MAC protocol. It also achieves a 46% decrease in packet drop probability, 79% increase in system throughput, and 25% decrease in mean packet delay.

**Keywords:** delay-tolerant mobile sensor networks; minimal probe frame; Wait to Send; Ready to Receive; queue management; priority; medium access control

---

## 1. Introduction

At present, most wireless sensor network (WSN) applications, *i.e.*, networks composed of large numbers of distributed sensor nodes that sample data, are limited to static sensor nodes. Such networks are good for natural environment monitoring and medical treatment, but their mechanisms are not sufficient for sensor applications [1,2] of sparse (low node density) networks that may appear in the future. In particular, they cannot deal with sensor node mobility. Motivated by the research on networks with low and intermittent connectivity, reference [3] proposes a delay-tolerant mobile sensor network (DTMSN) of wearable nodes that move with their carriers, gather information, and deliver it to a sink. The DTMSN distinguishes itself from conventional WSNs by its node mobility, sparse connectivity, delay tolerability [4-6] and highly dynamic network topology. Its sparse node density results in poor network connectivity [7]. Moreover, DTMSN possesses high delay tolerance and is affected by changes in queue size. These characteristics pose the following challenges for the design of MAC protocols.

- Power constraints: energy efficiency is often a major concern in WSNs due to the nodes' limited batteries. DTMSN sensor nodes must be conveniently portable. That means they need to have a most small button battery, a very limited energy source. Furthermore, the dynamic network topology reduces data transfer success rate, which in turn increases energy consumption. Therefore, an effective protocol mechanism to make efficient use of the limited energy of the nodes and thus extend the lifetime of the network is essential.
- Data delivery scheme: because of the sparse node density and high dynamic network topology of DTMSNs, mobile sensors are intermittently connected. This calls for the utmost use of the temporarily available communication links [3]. The data delivery scheme is the key to getting an adequate data transfer success rate.
- Queue-management strategy: in a DTMSN, data messages are delay-tolerant so that there is always a data queue in the memory of each sensor node ready for packet transmission. Hence, an appropriate queue management scheme is needed to sort the data messages in the queue, to determine which data message is to be sent when the sensor meets others or which message is to be dropped when the queue is full. Such a scheme should be able to avoid untargeted transfers, reduce packet transmission failures and meaningless retransmissions, and increase energy efficiency.

To reduce energy consumption, S-MAC [8] uses the RTS-CTS mechanism to wake up the neighbors of the sender and receiver after the packet is delivered; it avoids schedule misses and halves latency. T-MAC [9] improves on the design of S-MAC by shortening the period awake if the channel is idle. B-MAC [10] uses preamble sampling and eliminates synchronization to reduce energy consumption. X-MAC [11] and AS-MAC [12] maintain low power communications by using a shortened preamble approach. W-MAC [13] and CI-MAC [14] combine the MAC and routing functionalities to minimize the sleep delay and conserve energy in large-scale WSNs. The protocols above focusing on energy saving are usually used in WSNs. However, the network topology changes frequently in a mobile environment, and the data transmission fails if the topology information updates untimely. [15] put forward a dynamic queue management way to deal with the incoming data packets

but did not bring an optimal solution to achieve energy saving. Thus these MAC protocols do not work well in DTMSNs. To overcome their limitations, this paper proposes an energy-efficient MAC protocol using dynamic queue management, dubbed EQ-MAC. EQ-MAC uses a minimal probe frame (MPF) to complete neighbor discovery and updating and has a transfer mechanism initiated by the target receiver to valid successful data transmission and an effective data queue management strategy to deal with priority queuing and manage the data in the queue. MPF implements the neighbor detection with the shortest frame. Thanks to the MPF, the neighborhood table maintained by each node can be updated timely, and the node which is ready to send the data packet can detect the sink node as soon as possible. Each node sends an MPF periodically. A node that receives the MPF will update its neighbor table and check whether it has data to send to the target neighbor. If it has, it rapidly sends a WTS (Wait To Send) as a response to tell the sink the time needed to complete the data packet transfer and to tell other neighbors to go to sleep and save energy until the transfer finishes; then as a sender, it starts to transmit the data packet to the target node after receiving a RTR (Ready To Receive) from the sink. Otherwise, the node goes to sleep until the next waking period arrives. If a sensor node has a data queue waiting for sending, it prioritizes the dynamic queue data as either high-priority packets (H-Pkt) or low-priority packets (L-Pkt). If it meets two target sinks, it will process the H-Pkts before the L-Pkts.

The rest of the paper is organized as follows: Section 2 describes the design of the EQ-MAC protocol. Section 3 introduces the queuing model to analyze the protocol, and Section 4 describes the optimization for the following experiments. Typical performance results validated through the experiments are shown in Section 5. Section 6 presents the conclusions.

## 2. Design of EQ-MAC Protocol

EQ-MAC uses MPF to update the neighbor table so as to validate communications among sensor nodes in order to deal with poor network connectivity. To deal with the DTMSN's dynamic network topology, the target sink initiates the data transfer, thereby avoiding searches for active users, untargeted packets, and meaningless retransmissions. In addition, as a measure aimed at the data delay tolerance of DTMSN, EQ-MAC uses dynamic priority queuing management to handle the data packets in the queue. This management strategy is free from the packet blocking constraint.

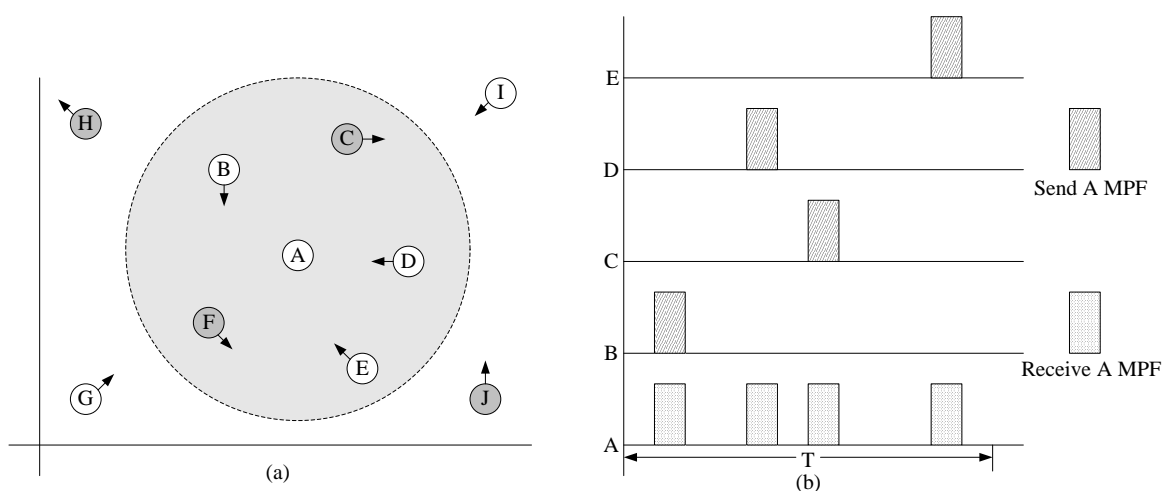
### 2.1. Neighbor Updating

In DTMSNs, asynchronous duty cycling is preferred over synchronous duty cycling. A synchronous duty cycling does not require nodes to share schedule information and require them only to stay awake long enough to sample the medium. It avoids overheads and excessive energy consumption. The application environment of EQ-MAC is delay tolerant, and data is not necessary to be sent immediately to the sink nodes. So there is no need to guarantee mobile nodes can wake up simultaneously when they are within each other's communication range. If they are awake, the sink node will send MPF to achieve synchronism. Otherwise they are waiting for the next appropriate occasion to exchange data and the nodes do not have to be wakened up to maintain synchronicity.

EQ-MAC employs a preambleless asynchronous scheme. Instead of a long preamble or a series of short preamble packets sent before the data packets, each node will wake up periodically and send

MPF. The awake neighborhood nodes will complete the neighbor discovery process once the MPF packets are received. In EQ-MAC's asynchronous duty cycle, each node broadcasts an MPF containing its own ID on schedule to notify all its neighbors of its identity. Upon receiving the MPF, awake neighbor nodes will update their neighbor tables immediately. As Figure 1(a) shows, nodes move in random directions, as denoted by the arrows. The rotundity field is the communication area of Node A; Nodes B, C, D, E and F can contact Node A, but Nodes G, H, I and J cannot. The dark nodes, *i.e.*, Nodes C, F, H, and J, are asleep during the fixed listening period of  $T$ . In Figure 1(b), once Node A wakes up, it starts listening to the channel. During the listening period, nodes in the communication range of Node A and that are awake broadcast MPFs. Nodes will detect the channel before sending MPF, and if there is communication in the channel, the nodes will back off for a random time. Otherwise, MPF will be sent immediately. Node A receives MPFs from its neighbor Nodes B, C, D, and E. Note that while Node C is asleep at the beginning of  $T$ , it wakes up later and broadcasts an MPF on schedule. Node F stays asleep during  $T$ , so Node A does not receive any information from it. When the listening period is over, Node A immediately updates its neighbor table according to the received MPFs.

**Figure 1.** Neighbor Updating.

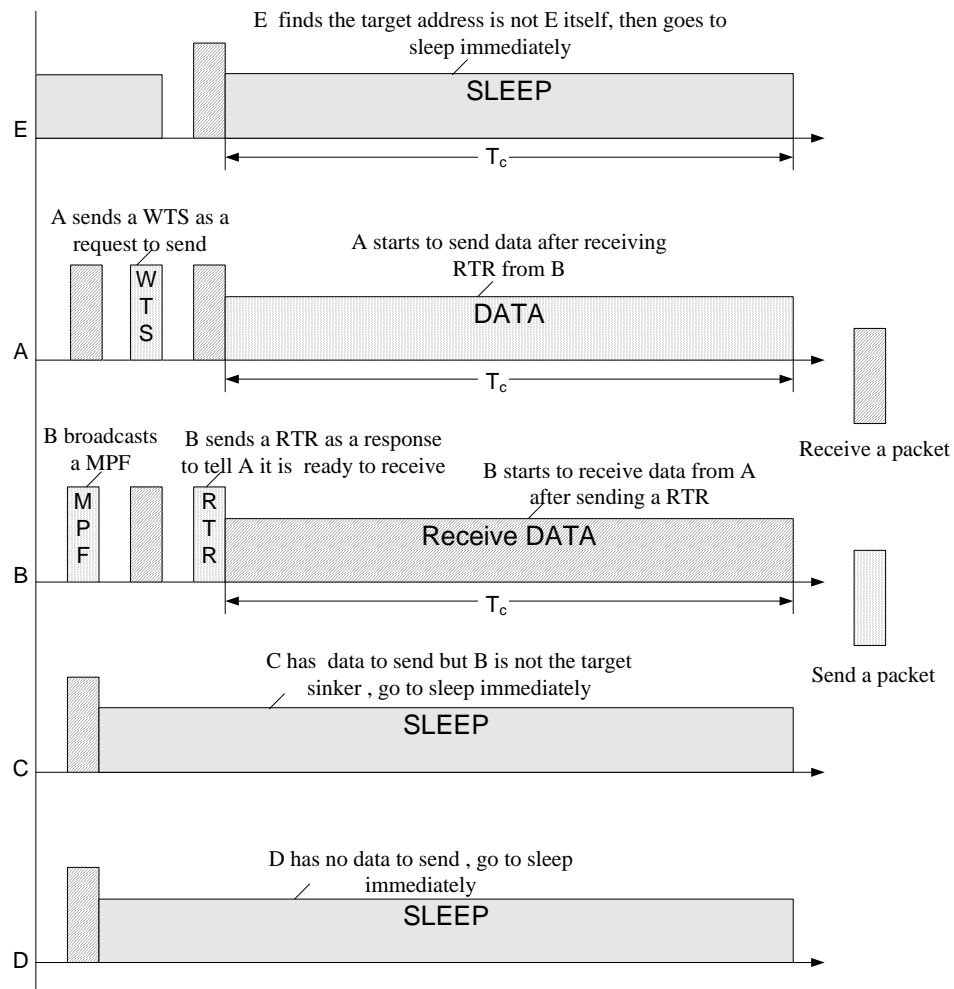


## 2.2. Data Delivery Initiated by Target Sink

Upon receiving the MPF, an awake neighbor node will check whether it has data waiting to be sent. If it does not, it will go to sleep. Otherwise, it will parse the MPF, get the ID and determine whether the target sink is the node sending the MPF. The neighbor node as a sender will send a WTS immediately after finding the target receiver, or else it will go to sleep. WTS is broadcasted so that other neighbor nodes will postpone their data transmission to avoid the hidden node problem. After receiving a WTS, the node that sent the MPF will know that there is data waiting for its acceptance. It then broadcasts an RTR as a response to inform the sender node waiting to start data transfers, since the target sink is ready to receive, and to notify the other awake neighbor nodes to go asleep until the whole communication is finished. The communication time is contained in the RTR frame. After three negotiations with the target receiver, the sender begins to transfer valid data.

Figure 2 illustrates sensor nodes running EQ-MAC in an asynchronous duty cycle. Node B, when it awakes, broadcasts an MPF to show its existence and inform other nodes of its waking state. The waking neighbor nodes (A, C, D) receive the MPF, get the ID of Node B contained in the MPF, and check to see if each node has data waiting to send. Node D has no data to be sent and thus goes to sleep immediately.

**Figure 2.** Data Delivery Initiated by Target Sink.



Although there is data waiting for Node C to send, the target sink is not Node B as its ID is not the same. Hence, Node C also goes to sleep. Node A, once it finds it has data waiting to be sent and the target receiver is just Node B, sends a WTS to tell Node B there is data waiting for its acceptance. After receiving the WTS from Node A, Node B sends an RTR as a response to inform Node A about starting the data transfer, which completes the data delivery scheme initiated by the target sink since the first time Node B sent an MPF as a request to receive data packets. RTR is also a message to tell other neighbor nodes to go asleep until the time interval  $T_c$  elapses. Like Node E, it is not woken till the interval in which RTR is broadcasted arrives. When receiving the RTR from Node B, Node E gets the time needed to complete the whole communication and the transmission address (TA) in the frame RTR. When the address of Node E does not match the TA, that is, Node B is not ready to receive data from Node E, it goes to sleep until the communication between Node A and Node B is finished. Node

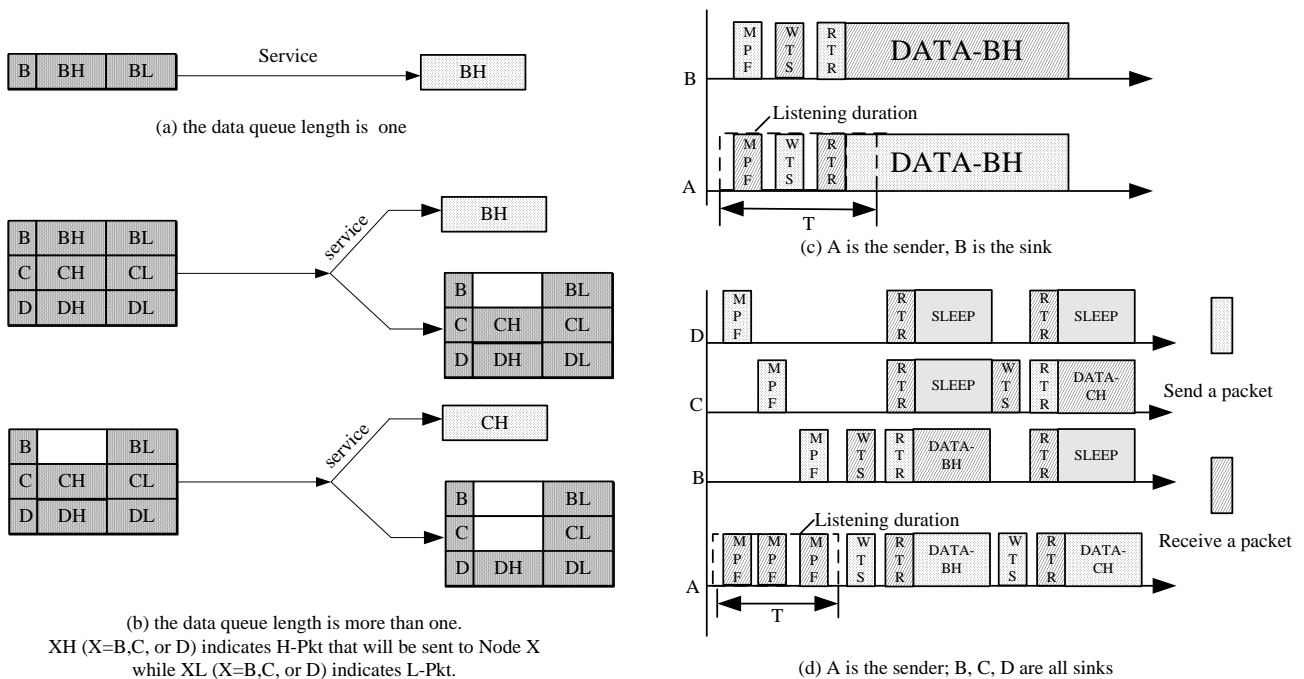
A begins to transmit data packets once it receives the RTR. At this point, the three-way handshake initiated by the target sink finishes and valid data transmission starts. By using data delivery initiated by the target receiver, reducing untargeted packet transfers, and avoiding meaningless data retransmissions which happen in highly dynamic network environments, EQ-MAC achieves significant energy saving and assures valid data transfers.

2.3. Queue Management Strategy Based on Priority

For the delay-tolerant characteristics of DTMSN, sensor nodes tend to have a data queue that contains data messages ready for transmission. A queue management strategy would appropriately process the data messages in the queue, sort the right data message to be sent when the sensor node meets another, and determine which data message should be dropped when the queue is full. In EQ-MAC, we employ a priority-based queue management scheme.

We distinguish two types of packet by their own priority. H-Pkts have priority over L-Pkts and are serviced first when both packet types are stored in the same data queue. In Figure 3, the number in the panel represents the priority of the data packet. The smaller the number is, the higher the priority the data packet is.

Figure 3. Data Queue.



In Figure 3(a), there is only one group of packets in the data queue which would be sent to Node B, and undoubtedly they can be served once the sensor node meets the target sink (Node B) and receives its MPF. The sender will first send H-Pkts (BH as shown in Figure 3(a)) for their higher priority. Figure 3(c) shows the data transmission process between Node A and B. The listening duration will end in advance when Node A meets Node B and detects Node B as the target. Then data transmission starts. When the data queue length is more than one, as shown in Figure 3(b), more than one group of data messages exist in the data queue. The data messages are grouped together according to the ID of

their target nodes. Each panel stands for one group of data messages and sorts the H-Pkts and L-Pkts that will be sent to the same sink node according to their own priority. The creation of the data queue and data packets sorting process can be handled efficiently, so the extra cost of the queue management in our MAC protocol implementation is negligible. Once the sensor meets the right receiver it prefers to send the H-Pkts.

For instance, when Node A has data to send, it first detects whether the data queue length is more than one. If it is only one, it will listen to the channel and wait for the MPF of the target sink (Node B). The listening process ends when Node A finds the target and receives an MPF from Node B (Figure 3(c)), and then data transfer starts. When the data queue length is more than one, the sensor needs to choose which data it should send first. As shown in Figure 3(b), there are three groups of data messages existed in the data queue of Node A, which need to be sent to Node B, C, and D. Each group sorts its data as H-Pkt and L-Pkt according to its priority. The data transmission process is shown in Figure 3(d). Node A first listens to the channel for a duration  $T$ , looking for the right target (Node B, C, and D) until  $T$  elapses. If it finds, it responds with a WTS immediately and starts data communication (between A and B). RTR broadcasts the communication time and other nodes (Node C and D) go to sleep during the duration. After the data transfers finish, the subsequent data transmission starts.

When the data queue is full, the sensor node will determine which group of data messages should be dropped. In EQ-MAC, the drop strategy is based on the priority. The higher priority the data message has, the later they will be dropped by the sensor. After new data has been gathered, the sensor finds there is no more space left for storing the incoming data messages and then compares the priority of the new data with the lowest priority of the messages in the data queue. If the new message has higher priority than the lowest priority in the original queue, the sensor will drop the lowest-priority data messages, push the new data into the data queue instead, prioritize the data messages in the data queue, and reorder them according to their priority. If the new message does not have higher priority, the sensor will drop the new incoming data messages.

### 3. Queuing Analysis

#### 3.1. Markov Chain

We developed a discrete-time Markov arrival process (MAP) priority queuing model to analyze the performance of the EQ-MAC protocol in the DTMSN environment. The queuing model is a preemptive priority queuing process [16]. Two conditions are assumed according to the preemptive priority queuing process in our paper. First, a communication preemptive priority queuing process, which means the ongoing communication will be interrupted and the new incoming data group will be preemptive to be sent to the sink if the contact between nodes exists and the incoming data messages have the highest priority; Second, data preemptive priority queuing process, that is to say, the incoming data will be preemptive to join in the data queue according to its priority without affecting the ongoing communication if there is any. When data packets are being sent between the sender and the sink, the priority of the new incoming data messages will be checked. If the incoming data has the higher priority than all the data in the queue including the sending packets and it is identified as the emergency data, the communication will be preempted by the new incoming emergency data. The

above is the communication preemptive priority queuing process. While there is no communication between nodes, the incoming data messages will join in the data queue according to its priority. Another situation is that, though the sender is contacting with the sink, the incoming data has the higher priority than all the data left in the queue but the lower (or the same) priority than the sending packets, or it is not emergency enough to stop the ongoing contact, it will also join in the data queue according to its priority and does not affect the ongoing communication. The two preemptive processes are suitable for the different conditions when meeting the incoming data of different priority. This queuing model can deal with the dynamic queue management of DTMSN effectively. Packet arrival is assumed to follow a Poisson process, a simple case of MAP in which  $\lambda_1$  and  $\lambda_2$  are the probabilities of arrival for H-Pkts and L-Pkts, respectively. The probability density function [17] is:

$$P_k(t) = \lambda_k e^{-\lambda_k t} \quad (t \geq 0, k = 1 \text{ or } 2) \quad (1)$$

The service time (*i.e.*, the time elapsed when sending the data packet, which is proportional to the length of the packet) distributions of the packets follow an i.i.d. negative exponential distribution, where  $\mu_1$  and  $\mu_2$  are the service rates for H-Pkts and L-Pkts, respectively:

$$P_k(s) = \mu_k e^{-\mu_k s} \quad (s \geq 0, k = 1 \text{ or } 2) \quad (2)$$

The number of H-Pkts in the data queue is  $i$ , and the number of L-Pkts is  $j$ . Let  $p(i, j; t)$  is the probability of  $i$  H-Pkts and  $j$  L-Pkts in the data queue at  $t$ .

The system utilization factor is:

$$\rho_k = \frac{\lambda_k}{\mu_k} \quad (k = 1 \text{ or } 2) \quad (3)$$

Given  $\rho_1 + \rho_2 = \frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} < 1$ , the balance equation is:

$$\begin{cases} (\lambda_1 + \lambda_2)p(0,0) = \mu_1 p(1,0) + \mu_2 p(0,1) , \\ (\lambda_1 + \lambda_2 + \mu_1)p(i,0) = \lambda_1 p(i-1,0) + \mu_1 p(i+1,0) , \quad i>0, j=0 \\ (\lambda_1 + \lambda_2 + \mu_2)p(0,j) = \lambda_2 p(0,j-1) + \mu_1 p(1,j) + \mu_2 p(0,j+1) , \quad i=0, j>0, \\ (\lambda_1 + \lambda_2 + \mu_1)p(i,j) = \lambda_1 p(i-1,j) + \lambda_2 p(i,j-1) + \mu_1 p(i+1,j) , \quad i>0, j>0, \end{cases} \quad (4)$$

where  $p(i,j) = \lim_{t \rightarrow \infty} p(i,j;t)$  ( $i,j \geq 0$ ).

To solve (4), we introduce the generating function:

$$\psi_i(z) = \sum_{j=0}^{\infty} p(i,j)z^j, \quad |z| < 1, i \geq 0 \quad (5)$$

$$\psi(u,z) = \sum_{i=0}^{\infty} \psi_i(z)u^i = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} p(i,j)u^i z^j \quad (|u| < 1, |z| < 1) \quad (6)$$

The state probability is:

$$p(i,j) = \frac{1}{i!j!} \left[ \frac{\partial^{i+j} \psi(u,z)}{\partial u^i \partial z^j} \right]_{u=z=0} \quad (7)$$

Let  $z \rightarrow 1$  and  $p_i$  be the probability when the number of H-Pkts is  $i$ . We get:



$$p_i = (1-\rho_1)\rho_1^i, \quad i \geq 0 \quad (8)$$

Let  $u \rightarrow 1$  and  $p_j$  be the probability when the number of L- Pkts is  $j$ . We get:

$$\sum_{j=0}^{\infty} j p_j = \frac{\rho_2}{(1-\rho_1-\rho_2)} \left[ 1 + \frac{\mu_2 \rho_1}{\mu_1 (1-\rho_1)} \right] \quad (9)$$

From Equation (8), we can easily obtain the average number of H-Pkts:

$$E[N_1] = \frac{\rho_1}{1-\rho_1} \quad (10)$$

From Equations (9) and (10), we get the average number of L-Pkts:

$$E[N_2] = \frac{\rho_2}{1-\rho_1-\rho_2} \left[ 1 + \frac{\mu_2}{\mu_1} E[N_1] \right] \quad (11)$$

### 3.2. Performance Measures

#### 3.2.1. Mean Packet Delay

The mean packet delay  $D$  is the average delay from the time data is sampled and queued to the time the packet is sent successfully. It can be obtained from Little's Law [17], as follows:

$$D = \frac{N}{\lambda} \quad (12)$$

where  $N$  is the average number of packets in the buffer of the sensor, and  $\lambda$  is the average probability of arrival for data packets.

From Equations (10), (11) and (12), we get:

$$D = \frac{E[N_1] + E[N_2]}{\lambda_1 + \lambda_2} = \frac{1}{(\lambda_1 + \lambda_2)} \left[ \frac{\rho_2}{1-\rho_1-\rho_2} \left[ 1 + \frac{\mu_2}{\mu_1} \times \frac{\rho_1}{1-\rho_1} \right] + \frac{\rho_1}{1-\rho_1} \right] \quad (13)$$

#### 3.2.2. Packet Drop Probability

The packet drop probability can be found by using the steady probability for the queue length of  $N_1$  H-Pkts (8), as follows:

$$p_d = \lambda_1^{-1} \times \left( 1 - \frac{\rho_1}{1+\rho_1} \right) \times (1 - p_{N_1}) = \frac{1}{\lambda_1(1+\rho_1)} \times [1 - (1-\rho_1)\rho_1^{E[N_1]}] \quad (14)$$

where  $E[N_1]$  is obtained from (10).

#### 3.2.3. Throughput

The system throughput [18] can be obtained from Equation (14):

$$S = \lambda_1 \times (1 - p_d) = \lambda_1 \times \left\{ 1 - \frac{1}{\lambda_1(1+\rho_1)} \times [1 - (1-\rho_1)\rho_1^{E[N_1]}] \right\} \quad (15)$$

#### 4. Optimization

We shall use an efficient mean estimation method to estimate the parameters related to EQ-MAC; that is, the mean value will be used to estimate the various values needed. The objective function is:

$$\varphi(t_i, t_r, t_l) = \varphi(\hat{t}_i, \hat{t}_r, \hat{t}_l) = \varphi(E[t_i], E[t_r], E[t_l]) \quad (16)$$

Here,  $t_i$  is the listening time from the moment one node is woken to the time it receives an MPF,  $t_r$  is a randomly chosen back-off time after receiving the MPF, and  $t_l$  is the listening time after the sink sends an MPF and before it receives the WTS.  $\hat{t}_i$ ,  $\hat{t}_r$  and  $\hat{t}_l$  are the estimated values of  $t_i$ ,  $t_r$ , and  $t_l$ .  $E[t_i]$ ,  $E[t_r]$ , and  $E[t_l]$  are the mean values of  $t_i$ ,  $t_r$ , and  $t_l$ .

We denote the average number that nodes contact in 1 second to be  $n$  and the mean number of data exchanges in each contact to be  $P$ . The time for checking the idle channel is  $t_c$ .  $L_p$  is the data packet length for one contact.  $L_M$ ,  $L_W$ , and  $L_R$  are the control packet lengths of MPF, WTS, and RTR.  $C_r$ ,  $C_t$ , and  $C_s$  are the average consumed current when nodes are receiving packets, transmitting packets, and asleep. The data rate is  $r_d$  and the supply voltage is  $V$ .

The average energy consumed by nodes sending packets in 1 second is:

$$\begin{aligned} E_t &= \text{energy for idle listening} + \text{energy for listening during back-off time} \\ &+ \text{energy for receiving MPF and RTR} + \text{energy for sending data} + \text{energy for sending WTS} \\ &= nV[(E[t_i] + E[t_r] + \frac{P(L_M + L_R)}{L_p r_d})c_r + (\frac{P}{r_d} + \frac{PL_W}{L_p r_d})c_t] \end{aligned} \quad (17)$$

The average energy consumed by nodes receiving packets in 1 second is:

$$\begin{aligned} E_r &= \text{energy for detecting the channel} + \text{energy for listening after sending MPF and before} \\ &\text{receiving WTS} + \text{energy for receiving data} + \text{energy for receiving WTS} \\ &+ \text{energy for sending MPF} + \text{energy for sending RTR} \\ &= nV[(t_c + E[t_l] + \frac{P}{r_d} + \frac{PL_W}{L_p r_d})c_r + (\frac{L_M + L_R}{r_d})c_t] \end{aligned} \quad (18)$$

The average energy consumption when nodes are idle in 1 second is:

$$\begin{aligned} E_i &= (\text{idle listening times} - \text{listening times with data exchange}) \times (\text{energy for channel detecting} + \\ &\text{energy for channel listening} + \text{energy consumed by sending MPF for neighbour updating}) \\ &= (\frac{1}{E[t_i]} - n)V[t_c + E[t_l]c_r + \frac{L_M c_t}{r_d}] \end{aligned} \quad (19)$$

The average energy consumption when nodes are asleep in 1 second is:

$$\begin{aligned} E_s &= \text{energy consumption during sleeping} \\ &= V\{1 - n[E[t_i] + E[t_r]] + \frac{P}{r_d}(1 + \frac{L_M + L_W + L_R}{L_p})\} - n[t_c + E[t_l] + \frac{P}{r_d}(1 + \frac{L_W}{L_p}) + \frac{L_M + L_R}{r_d}] \\ &\quad - (\frac{1}{E[t_i]} - n)(t_c + E[t_l] + \frac{L_M}{r_d})\}c_s \end{aligned} \quad (20)$$

The total average energy consumption of the network in 1 second is:

$$E = E_i + E_r + E_t + E_s \quad (21)$$

For energy efficiency,  $E$  should be minimized. We simply assume the mean value of the time is half of the time window and maximize the network lifetime by adjusting  $t_i$ ,  $t_r$ , and  $t_l$ :

$$\hat{t}_x = E[t_x] = \frac{t_x}{2} \quad (x = i, r, l) \quad (22)$$

From Equations (17–22), we get:

$$\frac{\partial E}{\partial t_i} = \frac{1}{2}nV(C_r - C_s) - \frac{1}{t_i^2}V\{(t_c + t_l)(C_r - C_s) + \frac{L_M}{r_d}(C_t - C_s)\} \quad (23)$$

To minimize  $E$ , let  $\frac{\partial E}{\partial t_i} = 0$ . Accordingly, we get:

$$t_i = \sqrt{\frac{2}{n}[t_c + t_l + \frac{L_M(c_t - c_s)}{r_d(c_r - c_s)}]} \quad (24)$$

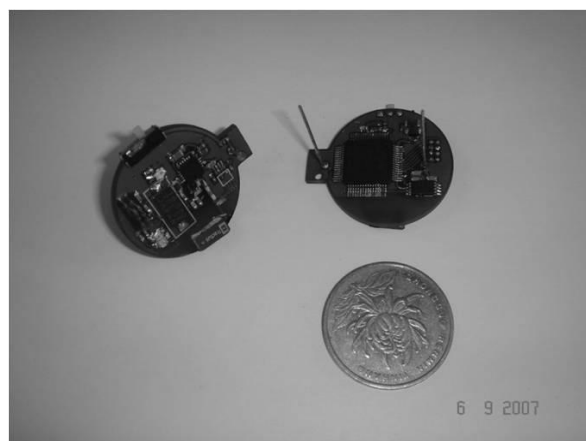
$t_l$  can be chosen according to empirical values and series of experimental results, thus  $t_i$  can be obtained from (24). By repeating the above process, groups of  $(t_l, t_i)$  can be recoded. And the best parameters can be fixed from the further experiments. Then  $E$  can be approximately minimized with the best values between  $t_i$  and  $t_l$ . Via the mean estimation method and optimized adjustment of experimental parameters, we can set up the network model and carry out successive experiments.

## 5. Performance Evaluation

### 5.1. Experimental Setup

To evaluate EQ-MAC, we performed a series of simple experiments on our experimental platform using ISC\_Motes, as shown in Figure 4.

**Figure 4.** ISC\_Mote.



The radio used by ISC\_Mote is the NRF24L01 chip, which has a maximum data rate of 2 Mbps and operates in the 2.4-GHz ISM band. The mote uses a TI MSP430F149 processor and fixes the temperature sensor DS620 to measure temperature. The experiment applications are based on the

tinyOS [19], which is an open source operating system, designed by UC Berkeley for WSNs. In EQ-MAC, nodes sleep in power-down mode. Moreover, they are in fact in Standby-I mode when they are idle listening [20].

We chose a playground about 10,000 m<sup>2</sup> in our campus as the experiment field. The communication distance of the mote is about 10 m. Twenty members of our institute were invited to join in the experiment. Each wore a sensor tag. The sensors sampled their carrier's human body temperature by DS620 while the participants moved around. Each participant could move randomly, which means he could move around within the playground or out of it as he wanted. The mote sampled the temperature every second, and stored the data into the queue. Data packets were grouped according to the ID of their sink, while each packet was tagged for a priority in every group. H-Pkts were generated by some of the participants who took strenuous exercise before the experiment and those whose temperature were slightly higher for the individual differences. If the average temperature was higher than 37.8 °C, the data packet was handled as H-Pkt, otherwise it was treated as L-Pkt. According to our experiment, the H-Pkts were about 25% in all the packets. When the sink was in the communication range of the sender and the handshake process was finished, packets would be sent to the sink. This experimental scene met with the network environment of DTMSN, where nodes were mobile, and data packets were delay-tolerant and could be gathered for a period, waiting for the suitable moment to be sent. We made a series of experiments by changing the node density and the packet arrival rate with each experiment for 3 hours. The experimental parameters are listed in Table 1.

**Table 1.** Experimental Parameters.

<i>Parameter</i>	<i>Value</i>
Transmission Current (mA)	23.4
Receive Current (mA)	25.8
Listen Current (uA)	148
Sleep Current (nA)	900
Supply Voltage (V)	3.0
Data Rate (kbps)	2,000
Battery Capacity (mAh)	580
Time for check the idle channel (ms)	50
Interval for sending MPF (ms)	500
Sleep period (ms)	500
Listening window (ms)	100
Length of MPF (bytes)	5
Length of RTR (bytes)	10

## 5.2. Numerical Results

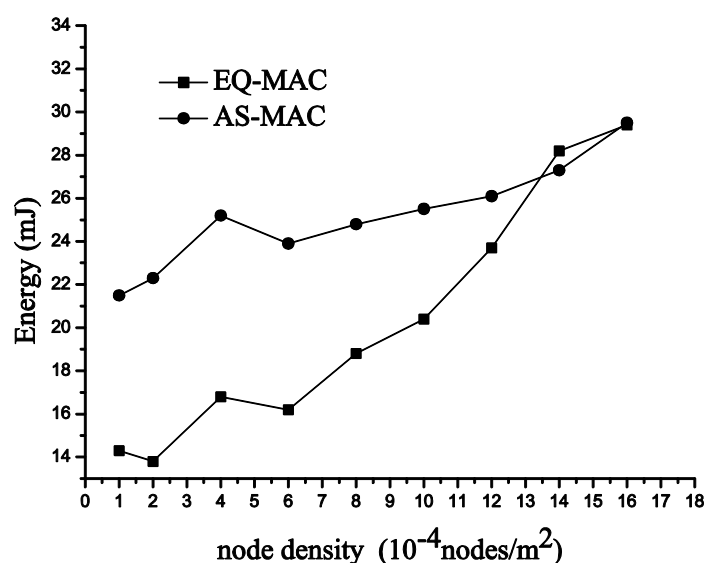
We implemented three MAC protocols in the experiment, EQ-MAC, nQ-MAC, and AS-MAC. EQ-MAC is the protocol with the dynamic queue management strategy as mentioned above. NQ-MAC follows EQ-MAC's receiver-initiated data transmission scheme but with no data priority, and the queue management without priority is used, thus the queuing model is the traditional queuing process.

The performance analysis can be easily obtained like the derivation in Section 3. AS-MAC is the protocol proposed in [12]. Numerical results are presented to show that EQ-MAC with receiver-initiated data transmission scheme outperforms AS-MAC which is commonly used in the traditional wireless sensor network in energy efficiency. The results also show that with the same receiver-initiated transmission scheme, EQ-MAC employing the dynamic queue management strategy can improve the probability of success in data transmission. The service rate of H-Pkt  $\mu_1$  was fixed at 0.9, while the service rate of L-Pkt  $\mu_2$  was set at 0.6.

### 5.2.1. Energy Efficiency

In a delay-tolerant and mobile network environment with a highly dynamic and sparse network topology, the average energy consumption in EQ-MAC visibly decreases compared with other MAC protocols in which the sender always initiates data transfers first. As Figure 5 shows, when the nodal density is less than 10, EQ-MAC yields a more efficient power saving in comparison with AS-MAC. The reason for this difference is their different data transmission mechanisms. In conventional MAC protocols like AS-MAC, data transfers are initiated by the sender. In a mobile and sparse network environment, this can easily lead to a transmission failure because the target node may be not in the contact area of the sender, and the consequently useless retransmission would increase energy consumption. But in EQ-MAC, it is the intended sink that initiates efficient data transfers through the neighbor updating process. This scheme assures successful data transmission, reduces the probability of data retransmission, and decreases transmissions of control packets; the result is higher energy efficiency.

**Figure 5.** Energy Consumption.

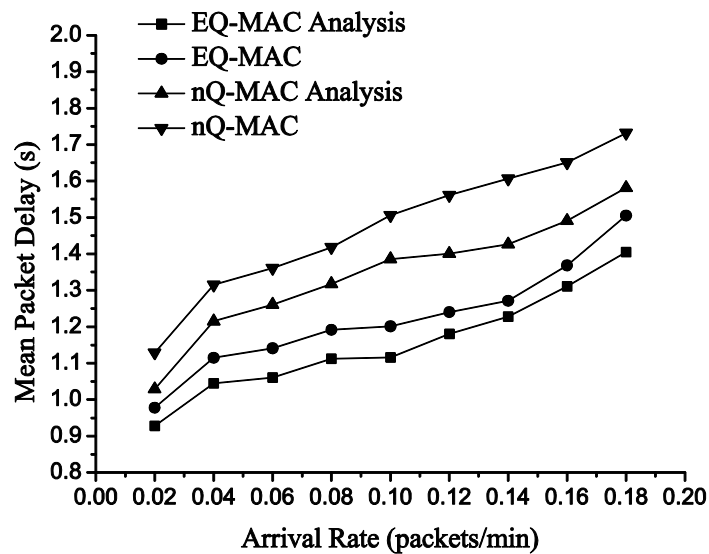


### 5.2.2. Mean Packet Delay

Figure 6 plots the mean packet delay. The mean packet delay for EQ-MAC is 25% less than that of the conventional MAC protocol. Such a significant difference in favor of EQ-MAC is attributed to the dynamic queue-management scheme based on priority. In the conventional protocol, packets are

transmitted according to their arrival sequence. On the other hand, the transmission sequence in EQ-MAC is based on the priority of packets, and we can dynamically manage the data queue. Thus, useful information can arrive in a more timely manner than in the non-queue-management strategy. Note that in the experiments, data packets were stored in memory and were sent or received by the hardware components of the ISC\_Mote. There were thus delays in the communications between the hardware components. This means the analytical results always appear to be better than the experimental ones.

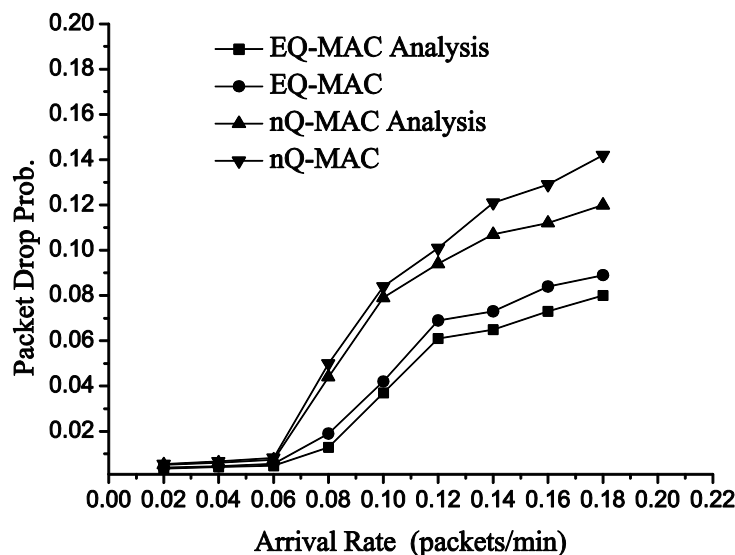
**Figure 6.** Mean Packet Delay vs. Arrival Rate.



### 5.2.3. Packet Drop Probability

Figure 7 plots the packet drop probability for H-Pkts against the effective arrival rate. We can see that the packet drop probability for H-Pkts in EQ-MAC becomes lower than that of the non-queue-management strategy as the arrival rate increases.

**Figure 7.** Packet Drop Prob. for H-Pkts vs. Arrival Rate.

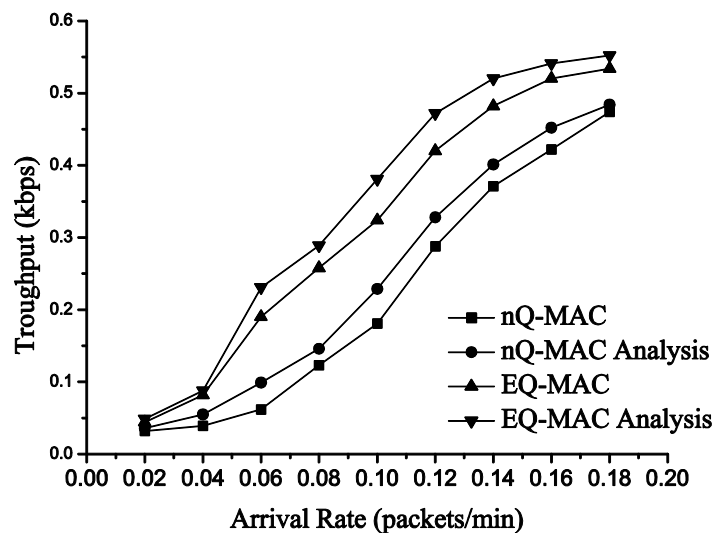


When the arrival rate is 0.12, EQ-MAC has a 46% decrease in packet loss compared with the non-queue-management strategy. Packet loss decreases since the dynamic queue-management scheme based on priority enables effective and rapid data processing and decreases blocked packets and accumulations of packets. Differences between theoretical analysis and experiments occur, for instance, when error bits occur during data transmission; the hardware components check the CRC, and when they find an error, they discard the packet. This is not accounted for in the theoretical analysis. Thus, there are always differences between the results of the analysis and the experiments.

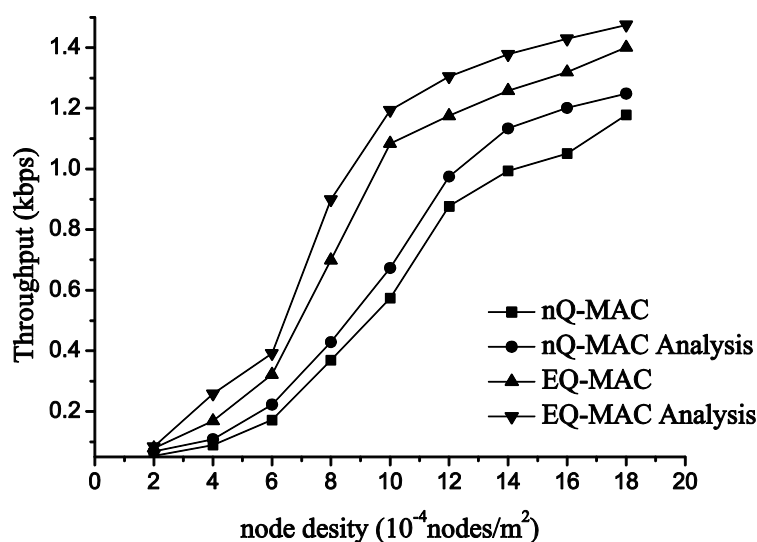
#### 5.2.4. Throughput

Figures 8 and 9 show the throughput for H-Pkts. When the network topology is sparse, EQ-MAC has better throughput because the process of data transfers initiated by the target sink ensures successful data transfers.

**Figure 8.** Throughput for H-Pkts vs. Arrival Rate.



**Figure 9.** Throughput for H-Pkts vs. Node Density.



Furthermore, the dynamic queue-management strategy increases the efficiency of H-Pkt transmission. When the node density is 10, the system throughput increases by 88.9% in comparison with the non-queue-management MAC protocol. When the arrival rate is 0.10, EQ-MAC has 79% increase in throughput compared with the non-queue-management MAC protocol. The results of the experiments and the theoretical analysis were somewhat different because the sending and receiving processes of the hardware were different from those of the theoretical analysis. For example, when data packets appear with high frequency, the wireless hardware module for sending and receiving data would have not enough time to deal with them or sufficient memory to accept and store some of them. Hence, some of these packets were discarded and could not be received in the experiment.

## 6. Conclusions

Nodal mobility and delay-tolerant data are two key characteristics in DTMSNs. Typically in medical treatment such as the epidemic tracking, infectious diseases are isolated and each wears a sensor tag for epidemic control. In this case nodes are mobile, and most sampling data is gathered for statistical analysis and only some urgent data is required to be sent immediately after it is sampled. In athletes' training surveillance, each athlete catches a sensor node to sample his body information like body temperature and heartbeat, which provides data for the coach to analyze and this information can be used to improve the athletes' training effort. These special mobile scenes with low node density and delay-tolerant data meet the application of DTMSN, and require an effective medium access control scheme to solve the problems of energy saving and data queue management.

Numerous schemes like duty cycling have been proposed for the medium access control layer of WSNs to avoid idle listening and reduce energy consumption. These schemes, however, operate poorly or do not have enough availability to work in the delay-tolerant mobile network environment of DTMSN without excessive energy consumption and data packet losses. We have proposed an energy-efficient MAC protocol using dynamic queue management called EQ-MAC as a way of dealing with the special characteristics of a highly dynamic and sparse topology and poor network connectivity in DTMSNs. Via data delivery initiated by the targeted sink, EQ-MAC guarantees that target gets the data intended for it, and it increases the data transfer rate and consumes less power in comparison with other protocols. Moreover, the targeted data delivery scheme reduces ineffective data packet retransmissions and useless transfers of control packets, leading to another obvious energy saving. EQ-MAC uses a dynamic queue management strategy based on priority to decrease the packet drop probability and mean packet delay. The strategy makes useful data reach the target terminal in a more timely manner. Experiments on a sensor platform verified that the dynamic strategy of EQ-MAC outperforms traditional MAC protocols not only in terms of energy consumption and mean packet delay but also in terms of packet drop probability and throughput in a DTMSN environment with highly dynamic and sparse topology.

## Acknowledgements

This research was supported by grants from the National Natural Science Foundation of China, Nos. 60872009, 60602016, and 61001081, the Hi-Tech Research and Development Program of China 863, Nos. 2007AA01Z428 and 2009AA01Z148, the Natural Science Foundation of Anhui,



Nos. ZD2008005-2, ZD200904, JK2009A013, and JK2009A025, the China Postdoctoral Science Foundation funded project, Nos. 20090460739, and the special grade of the financial support from China Postdoctoral Science Foundation, Nos. 201003328.

## References

1. Small, T.; Haas, Z.J. The Shared Wireless Infostation Model: A New Ad Hoc Networking Paradigm (or Where There Is a Whale, There Is a Way). In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*, Annapolis, MD, USA, 1–3 June 2003; pp. 233-244.
2. Small, T.; Haas, Z.J. Resource and Performance Tradeoffs in Delay-Tolerant Wireless Networks. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking*, Philadelphia, PA, USA, 22–26 August 2005; pp. 260-267.
3. Wang, Y.; Danq, H.; Wu, H. A Survey on Analytic Studies of Delay-Tolerant Mobile Sensor Networks: Research Articles. *Wirel. Commun. Mob. Comput.* **2007**, *7*, 1197-1208.
4. Wang, Y.; Wu, H. DFT-MSN: The Delay Fault Tolerant Mobile Sensor Network for Pervasive Information Gathering. In *Proceedings of the 25th International Annual Joint Conference of the IEEE Computer and Communications Societies*, Barcelona, Spain, 23–29 April 2006; pp. 1-12.
5. Wang, Y.; Wu, H. Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN): A New Paradigm for Pervasive Information Gathering. *IEEE Trans. Mob. Comput.* **2007**, *6*, 1021-1034.
6. Wang, Y.; Wu, H.; Lin, F.; Tzeng, N. Cross-Layer Protocol Design and Optimization for Delay/Fault-Tolerant Mobile Sensor Networks (DFT-MSN's). *IEEE J. Sel. Area. Comm.* **2008**, *26*, 809-819.
7. Choi, S.-C.; Lee, J.W.; Kim, Y. An Adaptive Mobility-Supporting MAC Protocol for Mobile Sensor Networks. In *Proceedings of the Vehicular Technology Conference*, Marina Bay, Singapore, 11–14 May 2008; pp. 168-172.
8. Ye, W.; Heidemann, J.; Estrin, D. An Energy Efficient Mac Protocol for Wireless Sensor Networks. In *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies*, New York, NY, USA, 23–27 June 2002; pp. 1567-1576.
9. Dam, T.V.; Langendoen, K. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, USA, 5–7 November 2003; pp. 171-180.
10. Polastre, J.; Hill, J.; Culler, D. Versatile Low Power Media Access for Wireless Sensor Networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, Baltimore, MD, USA, 3–5 November 2004; pp. 95-107.
11. Buettner, M.; Yee, G.V.; Anderson, E.; Han, R. X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, Boulder, CO, USA, 31 October–3 November 2006; pp. 307-320.
12. Liu, B.; Zhang, L.; Zhang, H. An Adaptive Schedule Medium Access Control for Wireless Sensor Networks. In *Proceedings of the Sixth International Conference on Networking*, Sainte-Luce, Martinique, France, 22–28 April 2007; pp. 12-18.

13. Niafar, S.; Shahhoseini, H.S. A New Sink Based Energy Efficient and Delay Sensitive MAC Protocol for Large Scale WSNs. In *Proceedings of the 12th International Conference on Symposium on Performance Evaluation of Computer & Telecommunication Systems*, Istanbul, Turkey, 13–16 July 2009; pp. 178-184.
14. Niafar, S.; Shahhoseini, H.S. Design a Cross-Layer Integrated MAC and Routing Protocol for Surveillance Systems Using WSNs. In *Proceeding of The International Workshop on Scalable Ad Hoc and Sensor Networks*, Saint Petersburg, Russia, 12–13 October 2009; pp. 1-6.
15. Li, J.; Qu, Y.G.; Li, Q.Y.; Zhao, B.H. A Queue Management MAC Protocol for Delay-Tolerant Mobile Sensor Networks. In *Proceedings of the 2nd International Conference on Advanced Computer Control*, Shenyang, China, 27–29 March 2010; pp. 426-430.
16. Fallahi, A.; Hossain, E. Distributed and Energy-Aware MAC for Differentiated Services Wireless Packet Networks: A General Queuing Analytical Framework. *IEEE Trans. Mob. Comput.* **2007**, *6*, 381-394.
17. Willig, A. *A Short Introduction to Queueing Theory*; Telecommunication Networks Group, Technical University Berlin: Berlin, Germany, July 1999.
18. Park, C.G.; Jung, H.S.; Han, D.H. Queueing Analysis of IEEE 802.11 MAC Protocol in Wireless LAN. In *Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies*, Morne, Mauritius, 23–29 April 2006; pp. 139-139.
19. *Tinyos Home Page*; Available online: <http://www.tinyos.net> (accessed on 25 June 2010).
20. *Nordic nRF24L01 Radios*; Available online: <http://www.nordicsemi.com> (accessed on 14 March 2010).

© 2011 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).