Minimize Waiting Time and Conserve Energy by Scheduling Transmissions in IEEE 802.11-based Ad Hoc Networks

Chih-Shun Hsu, Jang-Ping Sheu
Department of CSIE
National Central University
Chung-Li, 32054, Taiwan

Yu-Chee Tseng
Department of CSIE
National Chiao Tung University
Hsin-Chu, 30050, Taiwan

Abstract

The *mobile ad hoc network (MANET)* has attracted lots of attention recently. Most of the researches assume that every mobile host in the $MANET$ uses a fixed data rate and follows a *distributed coordination function (DCF)* to transmit messages. As we know that none of the research has combined multiple data rates and transmission scheduling to minimize waiting time and conserve energy for a $MANET$ with power-saving (PS) mode hosts. IEEE 802.11 has already supported multiple data transmission rate. However, how to decide the transmission rate is still an open question. Here, we propose a data rate selection protocol to select the best available data rate to transmit messages. After the data transmission rate has been selected, we can schedule each transmission according to the data transmission rate and the packet size. Our goal is to minimize the average waiting time of each transmission and thus the PS hosts can switch back to power-saving mode as soon as possible. Therefore, we follow the shortest job first policy to let the transmission with shortest transmission time to access the channel first. Simulation results show that our scheduling protocol can achieve high packet delivery rate, reduce waiting time and conserve lots of energy.

Keywords: mobile ad-hoc network (MANET), multiple data rate, transmission scheduling.

1 Introduction

Due to the advance of communication technology, wireless communication devices become cheaper and more popular. Mobile communication and computing become more and more important. One wireless network architecture that has attracted lots of attention recently is the *mobile ad hoc network (MANET)*, which consists of mobile hosts only (without base stations).

Most of the previous researches assume that every mobile host in the $MANET$ transmits messages with a fixed data rate. However, IEEE 802.11b [1] can support four different data transmission rates (1 M bits/sec, 2M bits/sec, 5.5 M bits/sec, and 11 M bits/sec) and Lucent's $WaveLAN$ II device has already supported multiple data transmission rates [2]. Therefore, we shall allow every mobile host to adopt the most efficient data rate to transmit messages as long as the channel quality permits, yet, how to detect the channel quality and decide the most efficient data transmission rate are still open questions.

Lucent's $WaveLAN$ II device [2] selects the transmission rate according to how many acknowledgements the mobile host has successfully received. However, the selected data rate may not be suitable for current channel condition. To improve Lucent's protocol, a Receiver-Based Auto Rate ($RBAR$) protocol for multi-hop $MANET$s is proposed in [3]. In the $RBAR$ protocol, the receiver will make the final decision of the transmission rate while it is exchanging RTS/CTS packets. However, it needs to modify the RTS/CTS packet and the physical layer header. A multiple data rates protocol for an infrastructure wireless LAN is proposed in [4]. In each beacon interval, the access point (AP) will broadcast several sub-beacon frames, each of them with different data rates. The mobile host will select its uplink data rate according to which sub-beacon frame it can decode. However, this protocol requires the help of the AP. An adaptive multiple data rates protocol for IEEE 802.11-based Wireless LAN is proposed in [5]. The sender's transmission rate is selected dynamically according to the detected SNR of the previous transmission/reception. However, the information got from previous transmission/reception may be out of date.

Several transmission scheduling protocols for IEEE 802.11-based $MANET$s are proposed in [6, 7, 8, 9, 10, 11, 12]. These protocols scheduling transmissions either according to the message's QoS requirement [6, 10] or the network's traffic load [7, 8, 9, 11, 12]. None of them schedules transmissions according to the data rate and the packet size.

None of the research has combined multiple data rates and transmission scheduling to minimize the waiting time of each transmission. In a single-hop $MANET$ with some power-saving (PS) mode hosts,

0-7803-7661-7/03/$17.00

the PS host can switch back to PS mode as long as it has no further message to transmit or receive. Therefore, minimize the waiting time of each transmission is to reduce the idle time of PS hosts and thus conserve energy. We first propose a data rate selection protocol for IEEE 802.11-based single-hop $MANET$s. After the transmission rate has been selected, we then schedule the MAC layer transmissions according to the transmission time so that the waiting time of each transmission can be minimized.

According to the standard of IEEE 802.11, when a host switches to PS mode, it shall wake up periodically for a short period of time, named as $ATIM$ window, to see if there is any pending message for it. When a host wants to transmit a message to a PS host, the sender shall transmit an ad hoc traffic indication map ($ATIM$) to the receiver during the $ATIM$ window. After received the $ATIM$ frame, the receiver shall reply an ACK to the sender and keep awake until the pending message has been received. Therefore, the receiver can select the data transmission rate according to the SNR of the $ATIM$ frame transmitted by the sender and then attach the selected data rates to the ACK frame. After receiving an ACK from the receiver, the sender can schedule its transmission according to the selected data rate and packet size. The sender basically follows the shortest job first policy to schedule the transmission. Therefore, we shall assign a higher priority to the transmission with a shorter transmission time (or a higher data rate and a shorter packet size). We set the initial value of contention window and the backoff timer according to the number of contenders and the priority of the transmission, respectively, so that the higher priority transmission has the higher priority to access the channel and the number of collisions can be reduced. Simulation results show that the transmission scheduling protocols can achieve high packet delivery rate, reduce lots of waiting time and conserve lots of energy.

The rest of this paper is organized as follows. Preliminaries are given in section 2. Our data rate selection protocol is shown in Section 3. In Section 4, we describe our transmission scheduling protocol. Simulation results are presented in Section 5. Section 6 concludes this paper.

2 Preliminaries

2.1 System Model

In this paper, we intend to design a data rate selection and a MAC layer transmission scheduling protocols for IEEE 802.11-based single-hop $MANET$s. We assume that most of the hosts in the $MANET$ are in PS mode and different data rate has different effective communication range [3, 4]. In IEEE 802.11b, there are four different data transmission rates in the $MANET$. Figure 1 shows that the four different data rates have four different effective communication ranges. The effective transmission radius for the four different data transmission rates are: 30 meters for 11 M bits/sec, 60 meters for 5.5 M bits/sec, 100 meters for 2 M bits/sec, and 200 meters for 1 M bits/sec, respectively. The lower the data rate is, the larger the effective communication range is. Therefore, to inform all hosts in the $MANET$, the messages (including $ATIM$, ACK and RTS/CTS frames and the broadcast packet) must be transmitted with the lowest data rate. Host A shall broadcast packets with data rate 1 M bits/sec, and can transmit unicast packet to host C with data rate 5.5 M bits/sec.

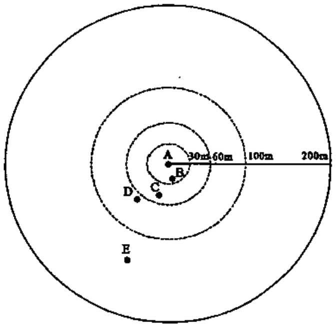

Figure 1: An example of the multiple data rates transmissions.

2.2 IEEE 802.11 MAC protocol

The IEEE 802.11 medium access (MAC) protocol [1] used in $MANET$s is the distributed coordination function (DCF) which is based on the *Carrier Sense Multiple Access with Collision Avoidance* ($CSMA/CA$) mechanism. When a mobile host wants to transmit frames, it first detects the status of the medium. If the medium is busy, the host will defer until the medium is idle for a period of time equal to $DIFS$ (DCF interframe space). After this $DIFS$ idle time, the host will generate a random backoff period, where *backoff time* = *Random*() $\times ST$. *Random*() is a random function, which is uniformly distributed between the interval $[0, CW]$ and ST is the length of a backoff time slot. The initial value of the CW is CW_{min}. When a host wants to send data, it first sense the medium. If the medium is idle for a period of time equal to $DIFS$, the backoff procedure will decrease the backoff time, otherwise, it will stop decreasing the backoff time. When the backoff timer expires, the host will transmit the frame. After the sender transmits the frame, if it is a broadcast, the receivers do nothing. Otherwise, if it is a unicast, the receiver will wait for a period of time equals to $SIFS$ (short interframe space, $SIFS < DIFS$) and then reply an Ack to the sender. If the sender does not receive an Ack from the receiver, the sender will double the size of its contention window and repeat the DCF procedure again.

2.3 Power-Saving Modes in IEEE 802.11-based $MANETs$

IEEE 802.11 [1] supports two power modes: *active* and *power-saving (PS)*. When a host switching its power mode, it must notify other hosts in the $MANET$. Therefore, each host will realize other hosts' power mode. Under an ad hoc network, PS hosts wake up periodically. The short interval that PS hosts wake up is called the *ATIM window*. It is assumed that hosts are fully connected and all synchronized, so the $ATIM$ windows of all PS hosts will start at about the same time. In the beginning of each $ATIM$ window, each mobile host will contend to send a beacon frame. Any successful beacon serves as the purpose of synchronizing mobile hosts' clocks. This beacon also inhibits other hosts from sending their beacons. To avoid collisions among beacons, a host should wait a random number of slots between 0 and $2 \times CW_{min} - 1$ before sending out its beacon.

After the beacon, a host with buffered unicast packets can send a direct $ATIM$ frame to each of its intended receivers in PS mode. $ATIM$ frames are also transmitted by contention based on the DCF access procedure. After transmitting an $ATIM$ frame, the mobile host shall remain awake for the entire remaining period. On reception of the $ATIM$ frame, the PS host should reply with an ACK and remains active for the remaining period. The buffered unicast packets should be sent based on the normal DCF access procedure after the $ATIM$ window finishes. If the sender doesn't receive an ACK, it should retry in the next $ATIM$ window. As for buffered broadcast packets, the $ATIM$ frames need not be acknowledged. Broadcast packets then can be sent based on contention after the $ATIM$ window finishes. If a mobile host is unable to transmit its $ATIM$ frame in the current $ATIM$ window or has extra buffered packets, it should retransmit $ATIM$s in the next $ATIM$ window. To protect PS hosts, only RTS, CTS, ACK, Beacon, and $ATIM$ frames can be transmitted during the $ATIM$ window.

Figure 2 shows an example, where host A wants to transmit a packet to host B. During the $ATIM$ window, an $ATIM$ frame is sent from A to B. In response, B will reply with an ACK. After the $ATIM$ window finishes, A can try to send out its data packet.

3 Our Data Rate Selection Protocol

Our data rate decision protocol is designed for $MANETs$ with hosts in PS mode. The receiver can select the best available data rate according to the quality of the received $ATIM$ frame. When both of the sender and the receiver are in active mode all the time, they select the data rate according to [3].

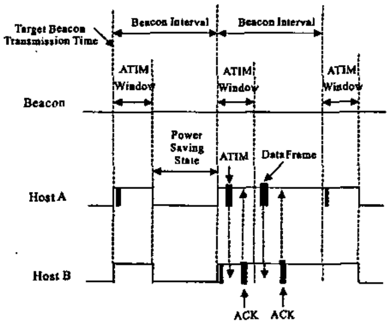

Figure 2: An example of unicasting in an ad hoc networks with PS hosts.

As mentioned in section 2.3, when the mobile hosts have switched to PS mode, they shall wake up periodically for a short period of time, named as the $ATIM$ window, to see if there is any pending message for them. If so, they shall keep active until the transmission is over, otherwise, it can switch back to PS mode when the $ATIM$ window is over. All the PS hosts' $ATIM$ windows start at about the same time. In the beginning of the $ATIM$ window, every PS hosts first follows the DCF procedure to transmit the beacon frame. After the beacon frame is successfully transmitted, any mobile host, which has buffered unicast packets to transmit (say H_s), then follows the DCF procedure to transmit the $ATIM$ frame to the receiver of the unicast packet (say H_r) with the lowest data rate (say 1 M bits/sec). On receiving the $ATIM$ frame from host H_s, host H_r compare the SNR of the $ATIM$ frame sent by host H_s with several predefined thresholds [13]. Assume that the SNR of the $ATIM$ frame sent by host H_s is S_s and there are $k+1$ predefined thresholds, denoted as T_1, T_2, …, T_{k+1}, where $T_i < T_{i+1}, T_1 = 0, T_{k+1} = \infty, i = 1 \ldots k$, and k predefined data rates, denoted as R_1, R_2, …, R_k, where $R_i < R_{i+1}$. If $T_i < S_s < T_{i+1}$, host H_s can transmit messages to host H_r with data rate R_i bits/sec.

Since the "duration" field of the ACK frame for the $ATIM$ frame is useless, we change the "duration" field to the "rate" field. However, the ACK frame for the data packet remains unchanged. After the transmission rate has been selected, host H_r will put the selected data rate into the "rate" field and then transmit the ACK frame to host H_s with the lowest data rate. After receive the ACK frame from host H_r, host H_s can schedule its transmission according to the selected data rate and the packet size. As for the broadcast packet, since all the mobile hosts in the $MANET$ are the receivers, the broadcast packet shall be transmitted with the lowest data rate.

4 Our Transmission Scheduling Protocol

We schedule the transmissions according to the transmission time. To minimize the average waiting time of each transmission and avoid the escort effect (that is the long job may occupy the channel for a long time and other jobs may have to escort the long job until it releases the channel.), we basically follows the shortest job first policy to schedule the transmission. Therefore, we shall let the transmission with shorter transmission time have the higher priority to access the channel. Although the broadcast packet must be transmitted with the lowest data rate, yet, the broadcast packet is small, all the hosts in the $MANET$ are the receivers of the broadcast packet and all the PS hosts can not switch back to PS mode until all broadcast packets have been transmitted. Therefore, we shall let the broadcast packet has the highest priority to access the channel so that the PS host can switch back to PS mode as soon as possible. If both of the sender and receiver are active all the time, they will wait until all the PS hosts' transmissions are over and then start their transmissions, since they are always active.

To guarantee that the transmission with the highest priority can access the channel first, we set the transmission's backoff timer according to the priority. The transmission with the highest priority will have the shortest backoff time and thus can access the channel first. At first, only the transmissions with the highest priority will contend to access the channel. After the transmissions with the highest priority are over, the transmissions with the second highest priority then contend to access the channel, and so on. Therefore, only the transmissions with the same priority will contend to access the channel at the same time.

In the following subsections, we first show how we set the priority of each transmission and calculate the number of contenders for each priority, and then we set the initial value of contention window (C_{min}) according to the number of contenders so that the waiting time of each transmission can be minimized and lots of collisions can be avoided. Finally, we set the transmission's backoff timer according to its priority so that the transmission with the highest priority can access the channel first.

4.1 Setting the Priority

As mentioned in Section 3, before really transmitting data packets, the sender should transmits an $ATIM$ frame to the receiver. For the convenience of calculating transmission time, the sender should attach the packet size to its $ATIM$ frame. After received the $ATIM$ frame, the receiver puts its selected data rate into the ACK frame and transmits it to the sender. Since the $MANET$ is single-hop and fully connected, each host can overhear other host's $ATIM$ and ACK frames during the $ATIM$ window and thus can calculate the transmission time and the number of contenders. As for the broadcast packet, we always use the lowest data rate to transmit the broadcast packet. The transmission time for broadcast packet is $\frac{packet\_size}{data\_rate}$ and unicast packet is $\frac{packet\_size+ACK\_frame}{data\_rate}+SIFS$. The broadcast packet always has the higher priority than the unicast packet to transmit on the channel. For the same type transmissions (broadcast or unicast), the transmission with the lowest transmission time has the hightest priority to access the channel. Therefore, each sender first divide the transmissions into three groups, one for broadcast, one for unicast of PS hosts and the other one for unicast of active hosts and then sort the transmissions in different groups according to their transmission times and assigns the priority according to the transmission's order. Each sender first assigns priority to the broadcast group and then assigns priority to the unicast group. Finally the sender will realize the priority of its own transmission. Each sender of the $MANET$ will maintain a table which records the transmission time and the number of contenders for each priority.

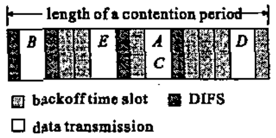

Figure 3: An example of 5 contenders transmit data in a contention period whose value of CW is set as 7

4.2 Setting the Initial Value of Contention Window

For any host H_i, given the number of its contenders, we will show how to set the initial value of H_i's contention window, so that the waiting time of each transmission can be minimized and lots of collisions can be avoided.

Without loss of generality, assume that the transmission priority of host H_i is p, there are n contenders and the initial value of CW is set as $m-1$. Under this condition, we will calculate the average interval between each successful transmission in this contention period. The shorter the interval is, the shorter the average waiting time is.

Figure 3 is an example of 5 contenders (hosts A, B, C, D, and E) transmit data in a contention period, whose value of CW is set as 7. Assume that there are 3 successful transmissions and 1 collision in the contention period. The total transmission and carrier sense time is $(3+1) \times (T_p + DIFS)$ and the total length of backoff time slots is $7 \times ST$, where T_p is the transmission time of a data packet and ST is the length of a backoff time slot. The length of the contention period (denoted as T_{cp}) is $(3+1) \times (T_p + DIFS) + 7 \times ST$ and the average interval between each successful transmission (denoted as I_{st}) is $T_{cp}/3$,. Therefore, to calculate the average interval between each successful transmission, first we need to evaluate the number of successful transmissions (denoted as $SUC(n,m)$) and the number of collisions (denoted as $COL(n,m)$) in the contention period, so that we can evaluate the length of the contention period and select the optimal initial value of CW.

For the convenience of calculating $SUC(n,m)$ and $COL(n,m)$, we first calculate the probability $pb(k)$, where $pb(k)$ is the probability that k ($k = 0, 1, \ldots, n$) hosts set its backoff timer as $(m-1) \times ST$ and the other $n-k$ hosts randomly set their backoff timers as $R \times ST$, where $R = 0, 1, \ldots, m-2$. Since each host has the same probability to set its backoff timer as any of the m different backoff window, the probability that a host set its backoff timer as $(m-1) \times ST$ is $\frac{1}{m}$ and the probability that a host broadcast set its backoff timer as $R \times ST$ is $\frac{m-1}{m}$. The number of combinations that randomly choose k hosts from n hosts is $C_k^n = \frac{n!}{(n-k)!k!}$. After analysis, we have $pb(k) = C_k^n (\frac{1}{m})^k (\frac{m-1}{m})^{n-k}$.

To calculate the expected value of $SUC(n,m)$ we need to calculate the probability and the average number of successful transmissions in each case. We can derive the recursive form of $SUC(n,m)$ according to the following analysis:

- **Case 1:** Assume that there are $(n-1)$ hosts randomly set their backoff timers as $R \times ST$, only one host set its backoff timer as $(m-1) \times ST$. The total number of hosts that transmit their messages successfully in this case is $SUC(n-1, m-1) + 1$ and the probability is $pb(1)$.
- **Case 2:** Assume that there are $(n-k)$ host randomly set their backoff timers as $R \times ST$ and k ($k \neq 1$) hosts set their backoff timers as $(m-1) \times ST$. There will be $SUC(n-k, m-1)$ hosts get their *ids* successfully in this case and the probability is $pb(k)$.

With the above analysis, we have
$SUC(n,m) = p(1)(SUC(n-1, m-1) + 1) + \sum_{k=0, k \neq 1}^{n} p(k) SUC(n-k, m-1)$

Figure 4 shows the expected number of successful transmissions ($SUC(n,m)$) in the first contention period. When the number of contenders (n) is fixed, as the size of contention window increases, the number of successful transmissions also increases.

With similar manner, we can derive a recursive from to calculate the expected number of collisions (denoted as $COL(n,m)$). We have $COL(n,m) = pb(0)COL(n, m-1) + pb(1)COL(n-1, m-1) + \sum_{k=2}^{n} pb(k)(COL(n-k, m-1) + 1)$.

Figure 5 shows the expected number of collisions ($COL(n,m)$) in the first contention period. When the number of contenders (n) is fixed, as the size of contention window increases, the number of collisions decreases.

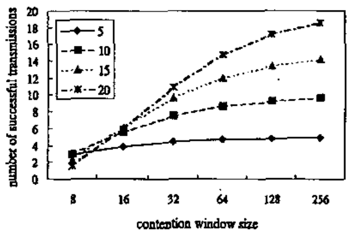

Figure 4: The expected number of successful transmissions ($SUC(n,m)$) in the first contention period

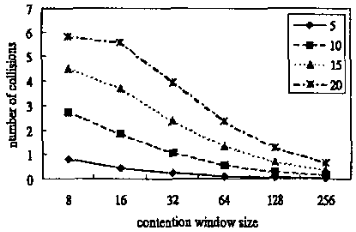

Figure 5: The expected number of collisions in the first contention period

When the number of contenders and the size of contention window are known, we can combine the previous two recursive forms to calculate the average interval between each successful transmission. The total transmission and carrier sense time is $SUC(n,m) + COL(n,m)) \times (T_p + DIFS)$ and the total length of backoff time slots is $(m-1) \times ST$. Therefore, the length of the contention period is $(SUC(n,m) + COL(n,m)) \times (T_p + DIFS) + (m-1) \times ST$ and the average interval between each successful transmission is $\frac{T_{cp}}{SUC(n,m)}$, where T_p is the transmission time of the transmissions with priority p. The shorter the interval is, the shorter the average waiting time is. Therefore, given the number of contenders we can calculate the average interval between each successful transmission for each m and find the optimal size of contention window (denoted as $CW_{opt}(n,p)$. Since the number of contenders may be very small, we tune the value of m from 2, 4, 8, 16, 32, 64, 128, to 256.

For example, assume that host H_i has 5 contenders (including itself), data transmission rate is 11 M bits/sec, the packet size is 1 K bytes, $ST = 20\mu s$, $SIFS = 10\mu s$, and $DIFS = 50\mu s$. Table 1 shows the evaluation results, we can see that when $m = 16$, the average interval between each successful transmission is smallest, therefore, host H_i sets its $CW_{opt}(5,p)$ as 15.

Table 1: Average interval (μs) between each successful transmission with 5 contenders

$n\backslash m$	2	4	8	16	32	64	128
5	5035	1530	1085	984	998	1103	1349

4.3 Setting the Backoff Timer

To guarantee that the transmission with the highest priority to access the channel first, we will set the backoff timer according to the priority of the transmission. Assume that the priority of host H_i's transmission is p, we will set the backoff timer for this transmission as follows:

$backoff\_timer = (\sum_{k=1}^{p-1} CW_{opt}(n_k, k) + Random()) \times ST$, where n_k is the number of contenders with priority k and $Random()$ is a random function, which is uniformly distributed between the interval $[0, CW_{opt}(n_p, p)]$.

This way we can not only guarantee that the transmission with the higher priority has the higher priority to access the channel, but also avoid the low priority transmissions contend with the high priority transmissions and avoid lots of collisions.

5 Simulation Results

To evaluate the performance of the proposed scheduling protocols, we have developed a simulator using C. In the simulations, we assume that the transmission radius is 200 meters, the length of a beacon interval is 200 ms, the length of an ATIM window is 20 ms, the unicast packet size is randomly selected between 512 $\sim$ 2048 bytes, and the broadcast packet size is randomly selected between 64 $\sim$ 512 bytes. The mobility part follows the random way point model, but the destination is within the single-hop $MANET$. The mobility is 10 meters/sec and the pause time is 30 seconds. In a single-rate protocol the transmission rate is 1 M bits/sec, while in a multiple-rate protocol the transmission rate is set according to Section 3. There are 40 hosts in the $MANET$, 80% of the hosts are in PS mode and the other 20% hosts are always active. The traffic load is tuning from 5 to 30 packets/sec with a Poisson distribution, 80% of the traffics are unicast and the others are broadcast. Each simulation lasts for 100 seconds. Each result is obtained from the average of 100 simulation runs.

Three performance metrics are used to evaluate our power-saving protocols:

- power consumption: the average power consumption for each mobile host in each second.

- waiting time: the average waiting time for each transmission. The waiting time is defined as the time after the ATIM window is over to the time the packet start transmitting.

Table 2: Power consumption parameters used in the simulation

Unicast send	$(454 + 1.9 \times L)/rate\ \mu W$
Broadcast send	$(266 + 1.9 \times L)/rate\ \mu W$
Unicast receive	$(356 + 0.5 \times L)/rate\ \mu W$
Broadcast receive	$(56 + 0.5 \times L)/rate\ \mu W$
Idle	$843\ \mu W/ms$
Doze	$27\ \mu W/ms$

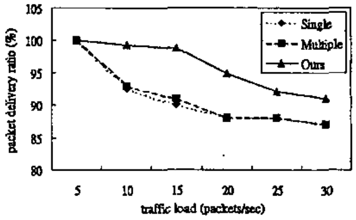

Figure 6: Packet delivery ratio

- packet delivery ratio: total number of received packet over total number of transmitted packet.

The power model in [14] is adopted, which is obtained by real experiments on Lucent WaveLAN cards. Table 2 summarizes the power consumption parameters used in our simulations, where L indicates the length of the packet and $rate$ indicates the data transmission rate of the packet. When sending a packet of the same size, unicast consumes more extra power than broadcast because it needs to send and receive extra control frames (RTS, CTS, and ACK); sending a packet with higher data rate consumes less power, because it uses less time to send the packet. The last two entries indicate the consumption when a host has no send/receive activity and is in the active mode and PS mode, respectively.

For simplicity, the single-rate transmission protocol without scheduling is denoted as $Single$, the multiple-rate transmission protocol without scheduling is denoted as $Multiple$, and the multiple-rate transmission protocol with scheduling is denoted as $Ours$. When without scheduling, each host follows the DCF procedure to transmit packets. Figure 6 shows that, the packet delivery ratio of our protocol are highest among the three protocols. Because in our protocol, each transmission is well scheduled to avoid contentions and collisions and thus can achieve higher delivery ratio. Figures 7 shows that, among the three protocols, our protocol waits least time to transmit a packet successfully. Since the delivery ratio is highest and the waiting time is shortest, as Figure 8 shows, our protocol consumes least power than the other two protocols. Among the three protocol, the single-rate protocol performs worst, because it takes a longer time to transmit a packet and thus consumes more power and takes a longer waiting time.

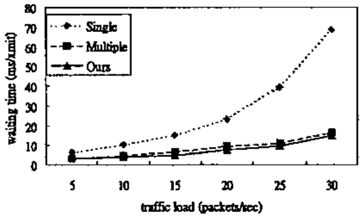

Figure 7: Average waiting time for each transmission

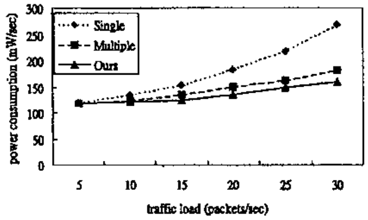

Figure 8: Average power consumption for each host in each second

6 Conclusions

In this paper, we propose a new data rate selection protocol and an efficient transmission scheduling protocol for a single-hop $MAMET$ with some PS hosts. The data rate is selected according to current transmission status and the priority of each transmission is set according to its data rate and packet length. Besides, broadcast has higher priority than unicast and PS host's transmission has higher priority than that of the always active host. Simulation results show that our scheduling protocols can achieve high packet delivery ratio, reduce lots of waiting time and conserve lots of energy.

References

[1] LAN MAN Standards Committee of the IEEE Computer Society, "IEEE Std 802.11-1999, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," *IEEE*, 1999.

[2] A. Kamerman and L. Monteban, "WaveLAN-II: A high-performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, pp. 118–133, 1997.

[3] Gavin Holland, Nitin Vaidya, and Paramvir Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks," *ACM SIGMOBILE*, pp. 236–250, 2001.

[4] Shiann-Tsong Sheu, Yang-Han Lee, and Meng-Hong Chen, "Providing Multiple Data Rates for Infrastructure Wireless Networks," *IEEE Global Telecommunications Conference*, 2001.

[5] Jean-Lien C. Wu, Hung-Huan Liu, and Yi-Jen Lung, "An Adaptive Multirate IEEE 802.11 Wireless LAN," *International Conference on Information Networking*, pp. 411–418, 2001.

[6] Imad Aad and Claude Castelluccia, "Differentiation mechanisms for IEEE 802.11," *IEEE INFOCOM*, pp. 209–218, 2001.

[7] Giuseppe Bianchi, Luigi Fratta, and Matteo Oliveri, "Performance Evaluation and Enhancement of the CSMA/CA MAC Protocol for 802.11 Wireless LAN," *International Symposium on Personal Indoor and Mobile Radio Communications*, pp. 392–396, 1996.

[8] Luciano Bononi, Marco Conti, and Lorenzo Donatiello, "Design and Performance Evaluation of a Distributed Contention Control (DCC) Mechanism for IEEE 802.11 WIreless Local Area Networks," *ACM workshop on Wireless Mobile Multimedia*, pp. 59–67, 1998.

[9] Frederico Cali, Marco Conti, and Enrico Gregori, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 785–799, 2000.

[10] Kanghee Kim, Seokjoo Shin, and Kiseon Kim, "A Novel MAC Scheme for Prioritized Services in IEEE 802.11a Wireless LAN," *IEEE International Conference on ATM and High Speed Intelligent Internet Symposium*, pp. 196–199, 2001.

[11] Hee-Cheol Lee and Ki-Jun Han, "Buffer overflow notification protocol at link level for wireless ad hoc networks," *Electronics Letters*, vol. 36, pp. 1899–1900, 2000.

[12] Marek Natkaniec and Andrzej R. Pach, "An Analysis of the Backoff Mechanism used in IEEE 802.11 Networks," *IEEE Symposium on Computers and Communications*, pp. 444–449, 2000.

[13] K. Balachandran, S. R. Kadaba, and S. Nanda, "Channel Quality Estimation and Rate Adaption for Cellular Mobile Radio," *IEEE Journal on Selected Areas in Comunications*, pp. 1244–1256, 1999.

[14] L. M. Feeney and M. Nilsson, "Investigating the Energy Consumption of Wireless Network Interface in an Ad Hoc Networking Environment," *IEEE INFOCOM*, pp. 1548–1557, 2001.