Wireless Networks 10, 61–69, 2004

A Priority MAC Protocol to Support Real-Time Traffic in Ad Hoc Networks *

JANG-PING SHEU ** and CHI-HSUN LIU
Department of Computer Science and Information Engineering, National Central University, Chung-Li, 320, Taiwan

SHIH-LIN WU
Department of Computer Science and Information Engineering, Chang-Gung University, Tao-Yuan, 333, Taiwan

YU-CHEE TSENG
Department of Computer Science and Information Engineering, National Chiao-Tung University, Hsin-Chu, 300, Taiwan

Abstract. Carrier sense multiple access and its variants have been widely used in mobile ad hoc networks. However, most existing access mechanisms cannot guarantee quality for real-time traffic. This paper presents a distributed medium access control protocol that provides multiple priority levels for stations to compete for the wireless channel. One common channel is assumed to be shared by all stations. Stations are assumed to be able to hear each other (i.e., the network is fully connected). The channel is accessed by stations according to their priorities, and for stations with the same priority, they send frames in a round robin manner. The channel access procedure is divided into three stages: priorities classification period, ID initialization period, and transmission period. Simulation results indicate that our protocol provides high channel utilization and bounded delays for real-time frames.

Keywords: carrier sense multiple access (CSMA), medium access control (MAC), mobile ad hoc network (MANET), quality-of-service (QoS), wireless communications

1. Introduction

The *mobile ad hoc network* (MANET) [9] has received a lot of attention recently. A MANET is formed by a cluster of mobile stations each equipped with a wireless network card. It can be quickly deployed without any established infrastructure or centralized administration. MANETs have applications in areas where infrastructure networks are difficult or impossible to be built (e.g., fleets on oceans, battle fields, festival field grounds, and historic sites).

Support of sufficient quality is critical for multimedia, real-time services in wireless communication systems. This work considers the medium access control (MAC) problem in MANETs. A MAC protocol should address the potential contention and collision problems among mobile hosts and at the same time utilize the communication bandwidth efficiently. MAC protocols can generally be divided into two categories: *centralized* and *distributed*. Centralized access schemes rely on an administration mechanism to coordinate the transmission of stations [2]. Examples include time division multiple access (TDMA), frequency division multiple access (FDMA), and code division multiple access (CDMA), where stations must reserve time slots, frequencies, and codes, respectively, to transmit their data. Polling is also a centralized scheme, where one common channel is shared by all stations but a station has right to use the channel only after it is polled by the coordinator. The PCF in IEEE 802.11 is one example.

Centralized access schemes are inappropriate for MANETs since a central administrator may not be available. Distributed access schemes, such as Aloha, CSMA, MACA [8], MACAW [1], and FAMA [6], may be more suitable for MANETs since they are mainly contention-based. The IEEE 802.11 standard provides two media access methods: the distributed coordination function (DCF) and point coordination function (PCF) [3]. The DCF protocol is designed for use in ad hoc networks and infrastructure wireless local area networks (WLANs) [4,7], while the PCF protocol is designed only for infrastructure WLANs. The fundamental access method in IEEE 802.11 is DCF, or known as *carrier sense multiple access with collision avoidance* (CSMA/CA), which supports asynchronous data transfer on a best-effort basis.

The IEEE 802.11 DCF mode, when applied to MANETs, does not provide a priority mechanism to support quality-of-service (QoS) transmissions. QoS guarantee is important for real-time traffic, such as video and voice, which requires time-bounded service and bandwidth guarantee, but stations still need to contend fairly with each other and with normal data traffic, such as text and e-mail. A simple priority scheme is proposed in [5], which is modified from the CSMA/CA protocol. The backoff scheme of the IEEE 802.11 is modified such that higher-priority traffics have the shorter backoff time. Access to medium is controlled through the use of different interframe space (IFS) intervals, such as SIFS, PIFS, and DIFS, between the transmission frames. A station using a shorter

---

* A preliminary version of this paper has appeared in the *International Conference of European Wireless, 2002* (*EW2002*).

** Corresponding author.
E-mail: sheujp@csie.ncu.edu.tw

IFS has a higher priority. This approach can not avoid some exceptional conditions. Collisions will occur whenever more than one station's timer expires simultaneously. The backoff window of a station depends on the number of retransmissions, so more backoff time is computed when more retransmissions occur. A station with higher-priority traffic may have longer backoff time than lower-priority ones. Also, the number of priority levels is limited to the number of different IFSs.

Sobrinho and Krishnakumar [12,13] proposed a priority protocol that is modified from the CSMA protocol to provide real-time access in a MANET. With this mechanism, a station with real-time traffic waits until a channel becomes idle for a PIFS period and then contends for the channel with pulses of energy, called the *black-burst* (BB). The period of BB is proportional to the time that the station has been waiting for the channel to become idle. After transmitting its BB, the station waits for an "observation" time to determine whether any other station is transmitting a longer BB. If the channel is perceived to be idle after this "observation" time, then the station begins to transmit its frame. However, a station will waste considerable channel bandwidth and energy to send BB for each outgoing frame.

A protocol called DBASE is proposed in [11]. This protocol also supports multimedia traffic in wireless ad hoc networks. Real-time traffic waits for a shorter IFS period than does non-real-time traffic to contend for the channel. The DBASE also uses the similar backoff scheme to contend for the channel, but the contention window's maximum size is smaller for real-time traffic than that for non-real-time traffic. A station that successfully obtains the channel will join a reservation table and do not need to contend the medium further throughout the whole session. DBASE uses a repetition interval, $D_{\max}$, which specifies the smallest maximal tolerance delay of all active real-time connections. Real-time stations in the reservation table will take turns to transmit frames during the $D_{\max}$ period. If the channel is idle for a DIFS period, non-real-time stations can contend for the channel until the $D_{\max}$ period expires. A small contention window for real-time traffic implies that much time is spent in contending the channel under a heavy load. In addition, DBASE assumes a long period of DIFS ($= 110\ \mu$s), which may degrade the channel utilization. The number of priority levels provided by DBASE is also limited to the number of different IFSs.

This paper presents a new MAC protocol that provides multiple priority levels. First, we adopt the BB mechanism [12,13] to separate higher-priority stations from lower-priority stations. By so doing, we guarantee that higher-priority frames are always transmitted earlier than lower-priority frames. Second, an ID initialization mechanism similar to that in [10] is used to schedule the transmission order of those stations with the same priority. These stations with the same priority then can transmit in a round robin manner. The advantage is that we can save lots of bandwidth and time because stations can transmit their frames consecutively according to their IDs without involving any contention resolution mechanism. Simulation results show that the performance of our protocol is superior to that of the IEEE 802.11 under the DCF mode.

The rest of this paper is organized as follows. Section 2 reviews a randomized initialization scheme that is used as a basis in our protocol. Section 3 presents the proposed priority MAC protocol. Simulation results are demonstrated in sections 4. Conclusions are drawn in section 5.

2. Review: A randomized initialization protocol

Our MAC protocol will be developed based on the *randomized initialization protocol* proposed in [10]. We review this protocol in this section. Minor modification will be made for use in our MAC protocol. The problem can be defined as follows: Given a set of n random stations, the purpose of the initialization protocol is to assign each of the stations a distinct ID number from 1 to n.

At this moment, let us assume for ease of presentation that each station has the collision detection (CD) capability. By CD capability, a host, when sending a packet, is able to detect whether there is a collision in this transmission by itself. (However, this is difficult in wireless radio transmission. Later on we will show how to relax this assumption.) With this assumption, a station can always determine the current channel status: silence, collision (transmissions from multiple stations), or busy (transmission from exactly one station).

The initialization protocol assumes that stations have no priority. The contest is fair. The basic idea is to construct a binary tree called a *contention tree*. From its position in the contention tree, a station can obtain a unique ID number. One single common channel is assumed, in which all stations will contend to send their request messages. A station which is able to send a request without collision is considered successfully obtaining an ID. The ith station successfully sending its request obtains an ID $= i$. If collision occurs, the station will flip a fair coin (with equal probability for head and tail). In case of head, the station will proceed to the left subtree (based on its current position in the contention tree) and continue to contend in the next round. In case of tail, the station will go to the right subtree and wait until all the stations in the left subtree obtain thier IDs, after which it can contend again.

Figure 1 shows a possible contention tree formed by five contenders/stations, A, B, C, D and E. In the beginning, all stations are assumed to stay in the root. In round 1, all stations will send their request messages simultaneously. Since this is a collision, each station flips a fair coin. Now suppose that A and B see heads and enter the left subtree, and C, D and E see tails and enter the right subtree. In round 2, A and B will continue to send their request messages. The result is a collision and A and B have to flip coins again. Let the result be heads again and thus both will find collision in round 3. Now suppose that A flips a head and B flips a tail. A will succeed in round 4 and obtains an ID of 1. This terminates the subtree rooted by A. B will send its request message in round 5, which will successfully get it an ID of 2. This also terminates the subtree rooted by B. In round 6, the channel

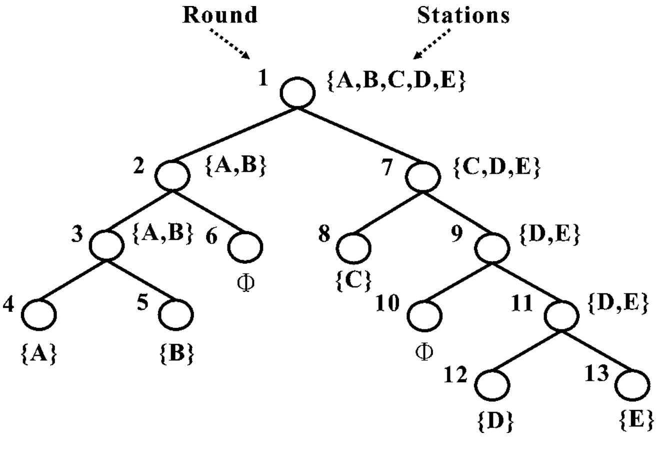

Figure 1. A contention tree of five stations.

will be silent since there is no station remaining on the left-hand side. The left subtree from the root is completed and in round 7 stations C, D and E will rejoin the contention. The process will repeat recursively until each station is assigned a unique ID. To summarize, a collision stands for an internal node in the contention tree, while a successful transmission or a silent status indicates a leaf node. After seeing the occurrences of all leaf nodes on the left subtree, stations on the right subtree can start their contention in the next round.

3. The proposed priority MAC protocol

In this section, we present our priority MAC protocol. Packets are prioritized into m levels. Higher levels mean higher priorities. The protocol will let all packets with higher priorities be transmitted earlier than those with lower priorities. The prioritizing process is done based on the *black burst* (BB) mechanism proposed in [12,13]. For stations with packets of the same priority, we will order these stations based on the randomized initialization protocol [10] discussed earlier. These stations will send their packets in a round-robin manner until all packets of the same priority are exhausted. Then stations with packets of the next priority level will join the competition again based on the BB mechanism.

Following the IEEE 802.11 standard, we assume that a station determines the medium to be idle by sensing the medium for a certain amount of inter-frame spacing (IFS). Three IFS intervals are used in our protocol: short IFS (SIFS), PCF IFS (PIFS), and DCF IFS (DIFS). We assume that PIFS $= 3 \times$ SIFS and DIFS $= 5 \times$ SIFS.

Our protocol consists of three basic mechanisms: (i) priority classification mechanism, (ii) ID initialization mechanism, and (iii) transmission mechanism. These mechanisms are introduced in the following subsections. They also partition the channel access into three periods: priority classification period, ID initialization period, and transmission period. At the end of this section, we will remark how to relax the assumption that stations have CD capability while they are sending.

3.1. Priority classification mechanism

Given a set of stations, this mechanism will classify them into m levels of priorities numbered from 1 to m. Higher numbers mean higher priority levels. Stations with higher-priority frames are allowed to contend the free channel first. Stations with lower-priority frames are blocked until all the higher-priority stations have completely transmitted their frames.

The black-burst (BB) scheme [12,13] is adopted here to distinguish the priorities of stations. Basically, BB is a jamming signal. The length of BB is proportional to the sending station's priority. Longer BB means higher priority. For a station of priority i, $1 \leqslant i \leqslant m$, it will send jamming signals for $BB_i = i \cdot t_{\text{unit}}$ amount of time, where t_{unit} is one BB unit time.

All stations that desire to contend for the channel will wait until the channel becomes idle for a DIFS period and then send out their BBs. A station that has completed its BB transmission will sense the channel status. If the channel is still blocked by BB signals, this means the station's priority is lower. It has to wait until the next DIFS period appears and then contend for the channel by sending BB again. A station that has exhausted its BB transmission and sensed the channel to be clean for a PIFS period can start to execute the ID initialization mechanism (refer to the next subsection). The illustration in figure 2 shows how our MAC protocol works.

The mechanism is formally presented below.

Step 1: A station that wants to transmit frames will first sense the status of the channel. If the channel is busy, the station will wait until it becomes idle for a DIFS period and then enter the priority classification period.

Step 2: Once the priority classification period starts, the station sends out BB signals to block the channel for BB_i period of time, where i is the priority level of the frames it intends to transmit.

Step 3: After sending its BB signals, the station has to sense the channel status. If there is no BB signal for a PIFS period, the station can enter the next ID initialization period (to be presented later). However, if the channel is still blocked by other stations' BB signals, it must keep on monitoring the channel until the channel becomes idle for a DIFS period and then goes back to Step 2 to contend the channel again.

For example, suppose that there are ten stations differentiated into three priority levels. Among them, five stations are of priority 3, three stations are of priority 2, and the rest are of priority 1. During the BB contention period, the five stations of priority 3 will transmit their BB signals for the longest time and then enter the ID initialization period. The other lower-priority stations must wait until all these five stations complete transmitting their frames before they can contend for the channel again.

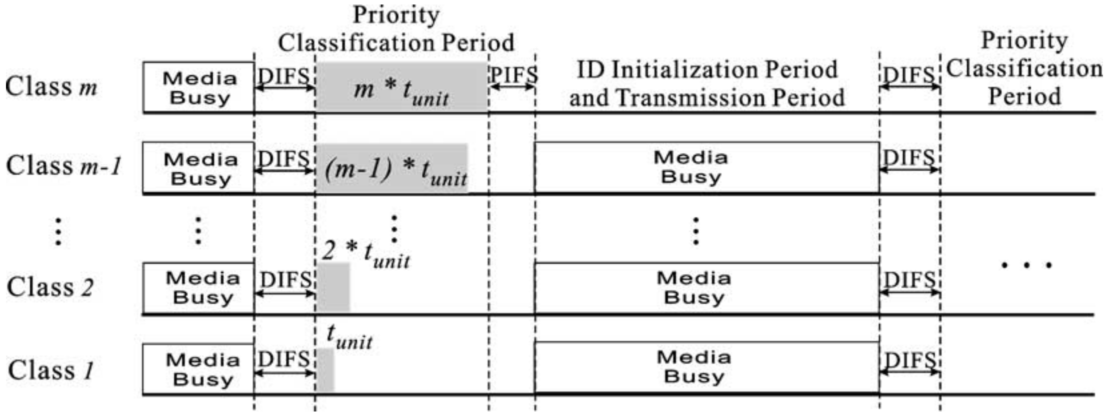

Figure 2. The priority classification mechanism of our protocol.

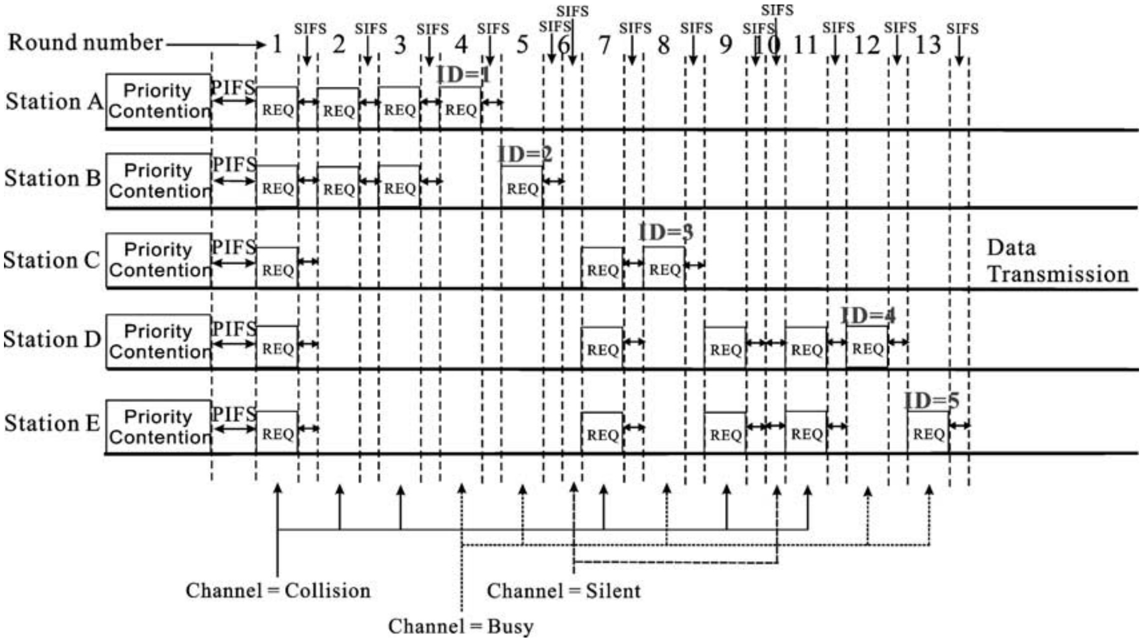

Figure 3. The ID initialization procedure based on the contention tree in figure 1.

3.2. *ID initialization mechanism*

After successfully passing the priority classification period, a station can enter the ID initialization period to execute the randomized initialization protocol. This mechanism is modified from the protocol in [10]. In this subsection, we still assume that stations possess the collision detection (CD) capability such that a sending station can detect whether its transmission encounters collision or not by itself.

Consider the example in figure 1. Figure 3 shows the corresponding ID initialization period. Stations' requests are denoted by REQs. These REQs are separated by SIFS periods. Whenever more than one REQ is sent, the collision status can be detected by all stations, including sending ones. The ID initialization period terminates after each station obtains a unique ID number.

In our protocol, the synchronization among stations is controlled by using different IFS intervals. After observing the channel remaining idle for a DIFS period, stations can undergo the priority classification procedure to contend for the channel. Stations with the highest priority wait for a PIFS period after the priority classification period to enter the ID initialization period. SIFS is used to separate REQ packets during the ID initialization period. Note that SIFS is also used in determining a silent channel (i.e., no station sending REQ when encountering a null subtree). For example, in rounds 6 and 10 of figure 1, there are no transmissions. Detecting the channel remaining silent for a SIFS period implies that this is a silent round, as illustrated in figure 3.

However, the above definition of IFS has danger in running into an erroneous state as shown below. The presence of contiguous silent rounds (each of length SIFS) during the ID initialization period may result in long silence in the channel. When the period of silence is equal to one DIFS, other lower-priority stations (if any) may incorrectly determine that they can start their priority classification period and start sending BB signals. This will interrupt the ID initialization period and all the previous efforts are wasted.

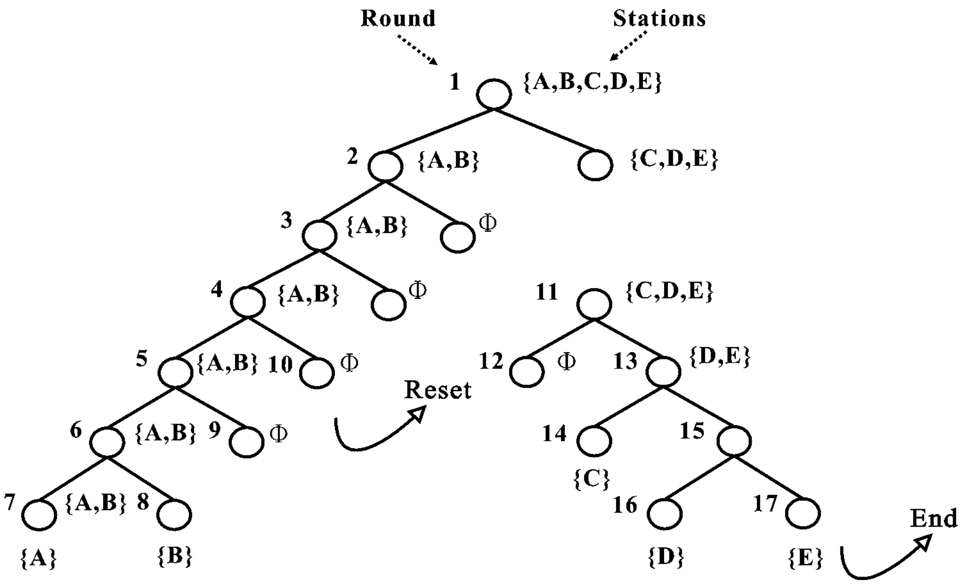

Figure 4. An example of the "reset" action: stations C, D, and E start a new contention tree in round 11.

This problem can be resolved as follows. A "reset" action can be taken after stations which have not obtained their IDs detecting two continuous idle rounds. These stations can restart a brand new contention tree. This can be done by having each such station send a REQ. All these stations now consider themselves staying in the root of a new contention tree. The process of constructing the new contention tree is same as above, but stations will contend for the next available ID. This is repeated until all stations obtain IDs. For example, in figure 4, on the left subtree, stations A and B do not resolve their collision until round 7. After round 8, there will be one SIFS. Then rounds 9 and 10 are silent, resulting in two more SIFS periods of silence. After that, in round 11, stations C, D, and E will form a new contention tree by sending their REQs. The second contention tree will assign each of C, D, and E a unique ID. Then these five stations can proceed to the next transmission period (to be presented in the next subsection).

The following description explains why the "reset" action should be taken after detecting two continuous silent rounds. Since we assume that DIFS $= 5 \times$ SIFS, we can tolerate at most four continuous silent SIFS periods. Consider the situation in figure 5. After round 13, all stations already have their IDs (however, no station is aware of this yet). One silent SIFS will appear after round 13 (to separate REQs, if any). This is followed by silent rounds 14 and 15. In round 16, a new contention tree is formed. However, since no station intends to obtain an ID, this round will be silent again. This terminates the second contention tree correctly, and the next transmission period can be started now. A total of 4 silent SIFS's have appeared. This explains why we cannot tolerate forming a new contention tree after detecting more than two continuous silent rounds.

For stations newly joining the network, they do not know the current status of the contention tree. One possibility is to let them wait until the next priority classification period. The other way is to attach sufficient information on REQ frames for them to join immediately. The later approach works as follows. A new station that enters the network and wishes to contend for a channel, will first sense the channel status. If the channel is not free, the station will monitor other stations' REQ frames. The station can join the ID initialization in the next round if it has the same priority. Several fields are included in each REQ frame for new stations to join the contention tree correctly, as shown in figure 6. The first four fields in REQ are similar to the design of RTS frames in IEEE 802.11. *TA* is the transmitter address, while *RA* is the receiver address. P represents the priority level of the sending station. L is the current number of collisions to be resolved, while l denotes when the station that can join the contention to send a REQ. N is the next available ID number yet to be assigned. Field *flag* contains an integer, which indicates the current number of continuous silent rounds that have been detected.

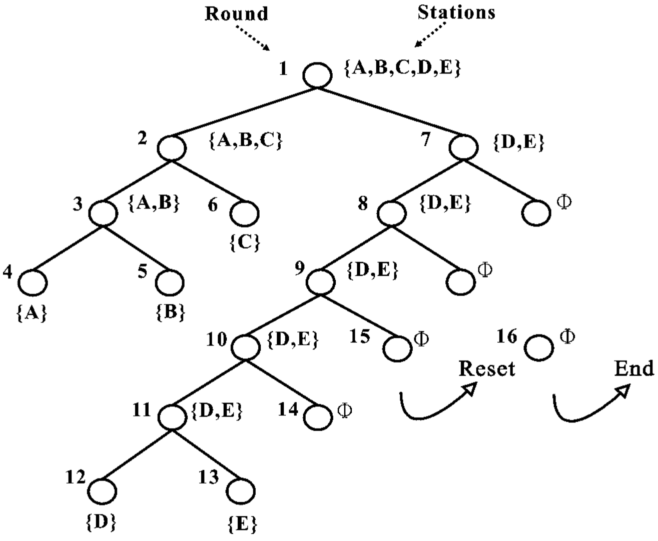

Figure 5. A scenario which causes four continuous silent SIFS's.

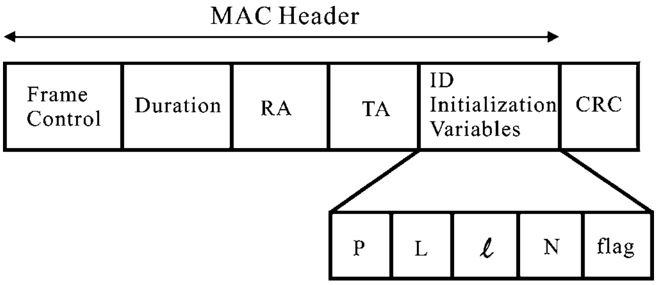

Figure 6. The REQ frame format.

The following steps show how these fields are calculated. Each station maintains for itself four variables, L, l, N, and *flag*. Initially, $L = l = N = 1$ and $\mathit{flag} = 0$. On sending REQ, these values are filled in the corresponding fields.

```
while $L \geqslant 1$ do
  1. For each station, if $L = l$, send out a REQ frame in the next round.
  2. Each station takes the following action, depending on the channel status.
    (a) status = "collision"
        Set $L \leftarrow L + 1$ and $\mathit{flag} \leftarrow 0$.
        If $L = l$, flip a fair coin. If the result is "heads", set $l \leftarrow L$.
    (b) status = "busy"
        The (only) station which broadcasts REQ sets $ID \leftarrow N$.
        All other stations set $N \leftarrow N + 1$, $L \leftarrow L - 1$, and $\mathit{flag} \leftarrow 0$.
    (c) status = "silent"
        All stations set $\mathit{flag} \leftarrow \mathit{flag} + 1$.
        If $\mathit{flag} = 1$, each station sets $L \leftarrow L - 1$.
        If $\mathit{flag} = 2$, each station without an ID number resets its local variables: $l \leftarrow 1$, $L \leftarrow 1$, and $\mathit{flag} \leftarrow 0$.
end while.
```

3.3. Transmission mechanism

After the ID initialization period, the stations who already own IDs can enter the transmission period. These stations will transmit their frames in a round-robin manner. The order that stations transmit can be in an ascending order of their ID numbers. In each turn, a station can send one data frame. Since stations do not necessarily have the same number of frames to be transmitted, a *piggyback* flag should be appended to each data frame. A $\mathit{piggyback} = 1$ means that the sending station still has more frames to be sent. Otherwise, the station should set $\mathit{piggyback} = 0$ and this station will be removed from the list of transmitters. These data frames should be separated by a SIFS period. Stations should also monitor the removal of any station from the list and keep track of their positions in the list.

3.4. ID initialization mechanism without using CD

The above ID initialization mechanism assumes that a sending station is able to detect immediately whether there is collision for its transmission. This assumption is impractical in wireless communication since its own signals will hit back. In the following, we propose a solution which does not rely on the CD capability.

We assume the availability of a leader election protocol, which can select a station as the leader of the network. (One simple approach is to let the station with the least MAC address serve as the leader. Other more distributed solutions, such as [10], are also possible.) The leader serves as the "virtual" collision detector in the network. The basic idea is to let the leader to send jamming signals with extra high energy in case that a collision is detected. The jamming signals' energy should be high enough such that a station can distinguish it from other regular transmissions. Note that by "regular transmission", we mean the transmission of a single packet or even multiple packets on the channel (the later case means a collision). This is made possible by sending the jamming signals with extra high energy.

The revised randomized initialization protocol works as follows. Note that in case of collision (with more than one REQ), we will waste one transmission round for transmitting jamming signals.

- For a station having no intention to send a REQ in a round, it will monitor the status of the round. If the round is silent or busy, it follows the original initialization protocol. If the round has collision, it will halt its execution in the next round (it is for sure that jamming signals will be sent in the next round, according to the next rule). After halting for one round, it resumes its execution of the original initialization protocol.
- For a station having intention to send a REQ in a round, it is unable to monitor the status of the current round. It must monitor the next round. If jamming signals are heard, it knows that its earlier transmission has collision. Then it follows the original initialization protocol in the subsequent round after the jamming signals (i.e., flip a fair coin and decide whether to enter the left or right subtree as usual). As long as no jamming signals are heard (i.e., the channel is silent or busy), the station should assume that its previous transmission is correct. Then it can calculate for itself an ID and exit the contention tree.

Below, let us assume that the leader does not participate in the initialization protocol. We shall prove that the above protocol is correct. We separate from three cases according to the status of the channel in one round. Clearly, if the channel is silent, all stations are aware of such. If the channel is busy (with exactly one sender), all stations are aware of such too, except the sender itself. It can tell this by monitoring the next round for jamming signals. This will not cause any problem since the sender (with successful REQ) is for sure to leave the contention tree. If the channel status is collision, all other stations except the senders themselves are aware of such too.

These non-senders will halt in the next round for the transmission of jamming signals. Those senders will also halt in the next round and, after hearing the jamming signals, will be able to resume the original initialization protocol after the jamming signals. As a result, excluding those halting rounds, the original initialization protocol is executed as usual.

Finally, we comment what should be done in case that the leader itself also intends to participate in the initialization protocol. If so, the leader should send jamming signals in the SIFS period after the first round. After the SIFS, the leader should send its REQ in the second round to confirm its intention. In this case, the leader's ID is always 1. All the other stations intending to participate in the initialization protocol simply halt in the second round and reset its contention tree as a new one. Then they resume the original initialization protocol in the third round, contending for the next ID, 2. Since the leader will not contend again, the protocol will run as normal.

4. Simulation results

This section evaluates the performance of our priority MAC protocol. The simulation model is built in a fully connected ad hoc network. The channel rate is assumed to be 11 Mbps and each frame size is 256 bytes. The following three traffic types are modeled.

- Pure Data: The arrival of data frames from a station follows a Poisson distribution.
- Voice: Such traffic follows the CBR (constant bit rate) model. The data rate of each voice stream is assumed to be 64 Kbps. The maximum tolerable delay for voice data is assumed to be 25 ms. Voice frames that are not transmitted within the maximum tolerable time are dropped.
- Video: Such traffic follows the VBR (variable bit rate) model. The maximum tolerable delay for video data is assumed to be 75 ms. The bit rate of each voice stream is exponentially distributed. Video frames that are not transmitted within the maximum tolerable delay are dropped.

Table 1 summarizes the parameters used in our simulation. We assume that video traffic has the highest priority, voice traffic has the second highest priority, and pure data traffic has the lowest priority. These types of traffics are mixed together with equal probability. Three performance measurements are used in our simulation:

- Throughput: the effective channel bandwidth for transmitting data/voice/video frames.
- Average Frame Delay: the average time from a frame's arrival until it is transmitted.
- Frame Loss Probability: the fraction of discarded voice/video frames due to the delay constraints.

We compare our protocol with the DCF protocol of the IEEE 802.11 standard. Two versions of our protocol are considered: one with CD and one without CD. Figure 7 compares the network throughput. After the offered load is beyond 3 frames/msec, differences can be seen. The throughput of our MAC protocol with CD can reach about 8 Mbps. Without CD, our protocol exhibits a slightly lower throughput. The throughput of the DCF protocol is about 6 Mbps.

Figure 8 shows the divide of throughput for each traffic type under our MAC protocol with CD. Under lighter loads, all types of traffic have equal change to be transmitted. When the load becomes heavier, data frames, which have the lowest priority, will be inhibited. The throughput of pure data traffic will gradually decline as the offered load increases. However,

Table 1
Simulation parameters.

Channel rate	11 Mbps
SIFS period	10 μs
PIFS period	30 μs
DIFS period	50 μs
Slot time	20 μs
Min. backoff window size	32
Max. backoff window size	1024
Length of control frame REQ	240 bit
Length of data frame	256 bytes
Voice source rate (CBR)	64 Kbps
Voice max. tolerable delay	25 ms
Video source maximum bit rate	420 Kbps
Video source average bit rate	239 Kbps
Video source minimum bit rate	120 Kbps
Video max. tolerable delay	75 ms

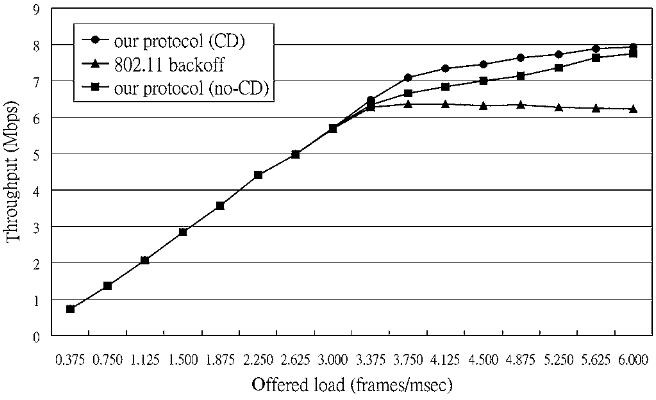

Figure 7. Comparison of throughput under different traffic loads.

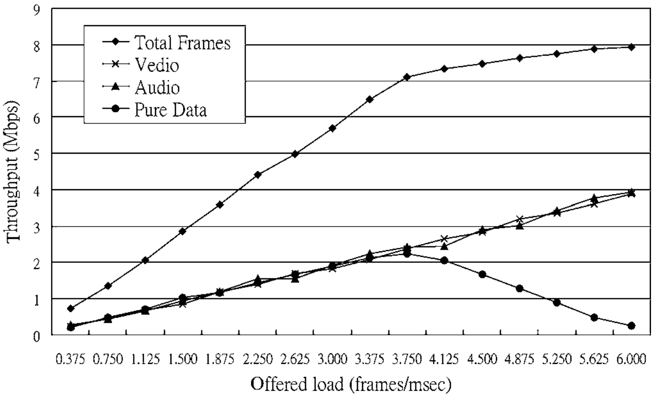

Figure 8. The divide of throughput into different traffic types for our protocol with CD.

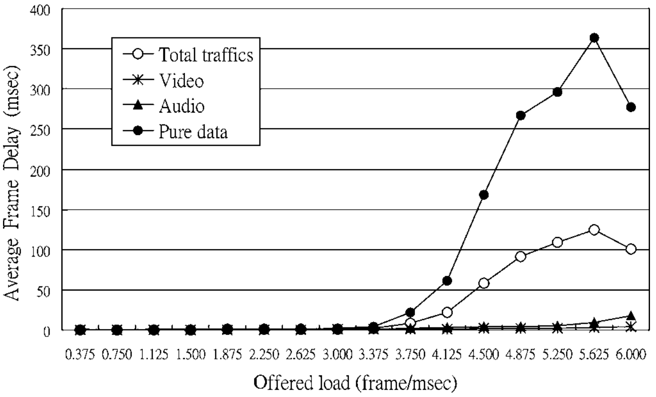

Figure 9. Average frame delays for different types of traffic for our protocol with CD.

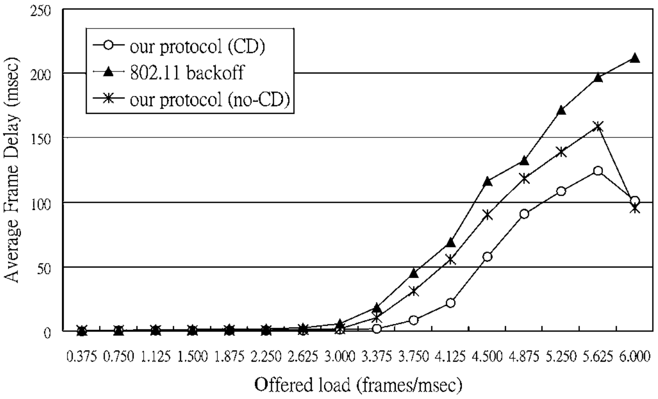

Figure 10. Comparison of average frame delay under different offered loads.

the throughput of real-time traffic is not affected. This shows that our protocol can effectively separate higher-priority traffic from lower-priority traffic.

Figure 9 shows the average delays for different types of frames under our MAC protocol with CD. Real-time frames do present much shorter delays than pure data frames do. Note that only frames that are successfully transmitted are counted in this measurement. When the load is very heavy, the overall average delay decreases because pure data frames have less chance to be transmitted, thus resulting in lower average delays. Figure 10 compares the average frame delay of our MAC protocol against the DCF protocol under different offered loads. The DCF protocol has the longest average delay, which is followed by our protocol without CD, and then by our protocol with CD. The advantage of our priority protocol is that as the load becomes heavier, the level of contention will not increase proportionally because some lower-priority stations are blocked.

Figure 11 shows the frame loss probabilities of voice and video packets for different protocols. The loss of the video traffic is close to zero as video traffic has the highest priority. The frame loss probability of our protocol is much less than that of the DCF protocol for both voice and video traffics. Therefore, our protocol is very suitable for transmitting real-time traffic.

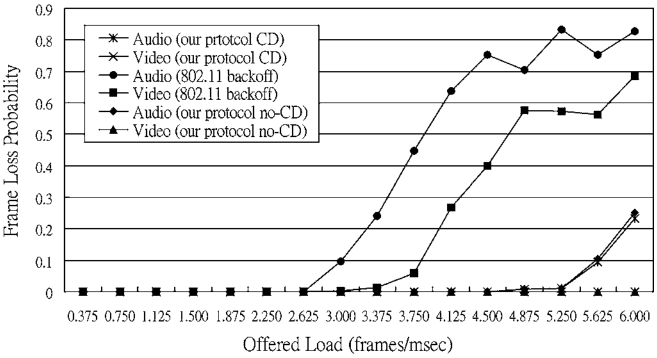

Figure 11. Comparison of frames loss probability of voice and video traffics under different offered loads.

5. Conclusions

The standard IEEE 802.11 DCF protocol does not provide a priority mechanism to support real-time traffic. This paper proposes a novel priority MAC protocol to support real-time traffic with delay constraint in a wireless ad hoc network. First, a priority classification mechanism is used to distinguish higher-priority stations from lower-priority stations. Second, stations with frames of the same priority are assigned consecutive ID numbers by a distributed mechanism. Finally, stations with frames of the same priority can transmit in a round-robin manner. Simulation results demonstrate that the proposed protocol is superior to the DCF protocol.

Acknowledgements

This work was supported in part by the MOE Program for Promoting Academic Excellent of Universities under the grant numbers A-91-H-FA07-1-4 and 89-E-FA04-1-4.

References

[1] V. Bharghavan, A. Demers, S. Shenker and L. Zhang, MACAW: A media access protocol for wireless LANs, in: *Proceedings of SIGCOMM '94* (1994) pp. 212–225.

[2] K.C. Chen, Medium access control of wireless LANs for mobile computing, IEEE Networks 8(5) (1994) 50–63.

[3] B.P. Crow, I. Widjaja, J.G. Kim and P.T. Sakai, Investigation of the IEEE 802.11 Medium Access Control (MAC) sublayer functions, in: *Proceedings of INFOCOM '97*, Vol. 1 (1997) 126–133.

[4] B.P. Crow, I. Widjaja, J.G. Kim and P.T. Sakai, IEEE 802.11 Wireless local area network, IEEE Communications Magazine (September 1997) pp. 116–126.

[5] D.J. Deng and R.S. Chang, A priority scheme for IEEE 802.11 DCF access method, IEICE Transactions on Communication (January 1999) 96–102.

[6] C.L. Fullmer and J.J. Garcia, Floor Acquisition Multiple Access (FAMA) for packet-radio networks, in: *Proceedings of SIGCOMM '95* (1995) pp. 262–273.

[7] *IEEE Std 802.11-1997: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications* (Institute of Electrical and Electronics Engineers, Inc., New York, 1997).

[8] P. Karn, MACA – A new channel access method for packet radio, in: *Proc. of ARRL/CRRL Amateur Radio Ninth Computer Networking Conf.* (1990) pp. 134–140.

[9] J.P. Macker and M.S. Corson, Mobile ad hoc networking and the IETF, Mobile Computing and Communications Reviews 2(3) (1998) 7–9.

[10] K. Nakano and S. Olariu, Randomized initialization protocols for ad hoc networks, IEEE Transactions on Parallel and Distributed System 11(7) (2000) 749–759.

[11] S.-T. Sheu and T.-F. Sheu, DBASE: A distributed bandwidth allocation/sharing/extension protocol for multimedia over IEEE 802.11 Ad Hoc Wireless LAN, in: *Proceedings of INFOCOM '01*, Vol. 3 (2001) pp. 1558–1567.

[12] J.L. Sobrinho and A.S. Krishnakumar, Real-time traffic over the IEEE 802.11 medium access control layer, Bell Labs Technical Journal (1996) 172–187.

[13] J.L. Sobrinho and A.S. Krishnakumar, Quality-of-service in ad hoc carrier sense multiple access wireless network, IEEE Journal on Selected Areas in Communications 17(8) (1999) 1353–1368.

Jang-Ping Sheu received the B.S. degree in computer science from Tamkang University, Taiwan, Republic of China, in 1981, and the M.S. and Ph.D. degrees in computer science from the National Tsing Hua University, Taiwan, Republic of China, in 1983 and 1987, respectively. He joined the faculty of the Department of Electrical Engineering, National Central University, Taiwan, Republic of China, as an Associate Professor in 1987. He is currently a Professor of the Department of Computer Science and Information Engineering, National Central University. He was a Chair of Department of Computer Science and Information Engineering, National Central University from August 1997 to July 1999. He was a visiting professor at the Department of Electrical and Computer Engineering, University of California, Irvine from July 1999 to April 2000. His current research interests include wireless communications, mobile computing, parallel processing, and distributed computing Systems. He was an associate editor of Journal of the Chinese Institute of Electrical Engineering, from August 1, 1996 to July 31, 2000. He was an associate editor of Journal of Information Science and Engineering from August 1, 1996 to July 31,2002. He is an associate editor of Journal of the Chinese Institute of Engineers. He was a Guest Editor of Special Issue for Wireless Communications and Mobil Computing Journal. He was a Program Chair of IEEE ICPADS'2002. He received the Distinguished Research Awards of the National Science Council of the Republic of China in 1993–1994, 1995–1996, and 1997–1998. Dr. Sheu is a senior member of the IEEE, a member of the ACM and Phi Tau Phi Society.
E-mail: sheujp@csie.ncu.edu.tw

Chi-Hsun Liu received the B.S. degree in computer science and information engineering from Chung Yuan Christian University, Chung-Li, Taiwan, R.O.C., in 1998, and the M.S. degree in computer science and information engineering from National Central University, Chung-Li, Taiwan, R.O.C., in 2001. His interests include computer network, wireless communication, multimedia networking and performance evaluation.

Shih-Lin Wu received the B.S. degree in computer science from Tamkang University, Taiwan, in June 1987 and the Ph.D. degree in computer science and information engineering from National Central University, Taiwan, in May 2001. Since August 2001, he joined the faculty of the Department of Electrical Engineering, Chang Gung University, Taiwan, as an Assistant Professor. His current research interests include mobile computing, wireless networks, network security, and distributed computing. Dr. Wu is a member of the Phi Tau Phi Society.
E-mail: slwu@mail.cgu.edu.tw

Yu-Chee Tseng received his B.S. and M.S. degrees in computer science from the National Taiwan University and the National Tsing-Hua University in 1985 and 1987, respectively. He worked for the D-LINK Inc. as an engineer in 1990. He obtained his Ph.D. in computer and information science from the Ohio State University in January of 1994. From 1994 to 1996, he was an Associate Professor at the Department of Computer Science, Chung-Hua University. He joined the Department of Computer Science and Information Engineering, National Central University in 1996, and has become a Full Professor since 1999. Since Aug. 2000, he has become a Full Professor at the Department of Computer Science and Information Engineering, National Chiao-Tung University, Taiwan. Dr. Tseng has served as a Program Committee Member in many international conferences, as a Program Chair in the *Wireless Networks and Mobile Computing Workshop*, in years 2000 and 2001, as an Associate Editor in *The Computer Journal*, and as a Guest Editor in *ACM Wireless Networks, IEEE Transactions on Computers, Journal of Internet Technology*, and *Wireless Communications and Mobile Computing*. His research interests include wireless communication, network security, parallel and distributed computing, and computer architecture. Dr. Tseng is a member of the IEEE Computer Society.
E-mail: yctseng@csie.nctu.edu.tw