QGMA: A New MAC Protocol for Supporting QoS in Wireless Local Area Networks [*]

Yi Ye, Chao-Ju Hou, and Ching-Chih Han
Department of Electrical Engineering
The Ohio State University
Columbus, OH 43210-1272
{*yiye,jhou,cchan*}*@ee.eng.ohio-state.edu*

Abstract

In this paper, we propose a novel MAC protocol, called Quality-of-service Guarantee Multiple Access (QGMA), in wireless local area networks to support the quality of service required by embedded real-time applications and multimedia-integrated distributed control. As compared to other existing MAC protocols, QGMA achieves high system utilization, and provides a deterministic bound on the connection establishment delay and temporal QoS for real-time applications. In particular, by having the base station handle resource arbitration and utilize different scheduling algorithms to arbitrate access to data slots on a wireless channel, mobile hosts (MHs) are relieved from the burden of coordinating all the ongoing activities, and moreover, different levels of temporal QoS can be easily achieved.

1. Introduction

Wireless communication has become an important technique for supporting emerging Personal Communications Service (PCS). In PCS, both traditional telephone service and other more advanced data applications, e.g., personal mail, audio/video, and multimedia-integrated real-time applications, are expected to be simultaneously supported. Data applications such as electronic mail and file transfer, require data accuracy, while multimedia-integrated real-time applications require that packet delay be less than or equal to a user-specified delay bound. An effective medium access control (MAC) protocol must be carefully designed to support different levels of QoS in PCS. The main intent of this paper is thus to design a MAC protocol, called *Quality-of-service Guarantee Multiple Access* (QGMA), that supports QoS for wireless local area networks (WLANs).

A wireless network is composed of a number of wireless local area networks (WLANs). A WLAN is the basic unit of a wireless network, and is also referred to as a *cell*. Each WLAN is composed of a base station (BS), a variable number of mobile hosts (MHs) and a fixed number of frequency channels on which the BS and MHs communicate. A BS is the central unit of the WLAN and is connected to one another to form a wired point-to-point backbone network. It also handles all the ongoing activities in the WLAN, and has the overall control of the system resources. Since the bandwidth available in a WLAN is limited, a MAC protocol is needed to efficiently coordinate the activities of the MHs that coexist in a WLAN. Several MAC protocols have been proposed for WLANs, e.g., Packet Reservation Multiple Access (PRMA) [8], Dynamic TDMA (D-TDMA) [11], Floor Acquisition Multiple Access (FAMA) [3], Resource Auction Multiple Access (RAMA) [1], and Dynamic Reservation Multiple Access (DRMA) [10], just to name a few.

Most of the MAC protocols proposed to use the reservation-based Time Division Multiplexing Access (TDMA) scheme or a variation thereof. In some of the proposed MAC protocols, dedicated bandwidth is assigned to MHs for their reservation requests, while in others contention-based approaches are used for reservation requests and all the reservation requests from MHs contend with one another for the bandwidth. One problem that arises in the dedicated bandwidth assignment approaches is inefficient use of bandwidth. On the other hand, contention-based approaches suffer from low utilization of reservation bandwidth under medium to heavy traffic loads. Also, due to the randomness nature of contention-based approaches, the time needed to successfully make a reservation request is probabilistically unbounded.

In this paper, we propose a novel MAC protocol, called QGMA, that eliminates the drawbacks of existing MAC protocols, achieves higher system utilization, and provides a deterministic bound on the time needed to make reservation requests. QGMA also supports, through the use of different scheduling algorithms by the BS, temporal QoS. As the BS in a WLAN usually has the overall control over all the system resources, QGMA delegates to BSs to arbitrate the assignment of data slots on a channel. The feature of supporting temporal QoS is especially desirable for real-time applications in which failure to transport data packets in a timely manner may depreciate the value of packets.

To present the key features of QGMA, we first discuss the base protocol that considers only the transmission from MHs to the BS and supports only one class of service. After presenting all the essential features, we then discuss how to extend the base protocol to make its functions more complete and flexible. To demonstrate the ability of QGMA in supporting temporal QoS, we describe how to incorporate a DCTS-based algorithm into QGMA. Finally, we validate the design of QGMA via simulation study.

The rest of the paper is organized as follows. In Section 2, we introduce the system model under consideration and the message model used to characterize real-time message streams with temporal QoS requirements. We present our base protocol in Section 3 and the extensions to the base protocol in Section 4. In Section 5, we demonstrate, after a brief introduction on the DCTS-based scheduler how to incorporate it into QGMA to support temporal QoS. We summarize several existing MAC protocols for WLANs in Section 6, and empirically evaluate

[*] The work reported in this paper was supported in part by the DARPA under Contract No. N66001-97-C-8526 and by the NSF under Grant No. NCR-9625064.

(via event-driven simulation) QGMA against the other MAC protocols in Section 7. The paper concludes with Section 8.

2. System and Message Models

2.1. System Model

We define a transmission pair to be a BS and one of the MHs in the cell. At any time, only one transmission pair is active on the channel; otherwise, a collision is said to occur and all the outstanding transmissions fail. (The capture effect, which allows collided transmissions to be recovered, is not considered in this paper).

There are two types of traffic in a cell: data packets and control packets. The system utilization, U, is defined as the ratio of the time used in (successfully) transmitting data traffic to the total time elapsed, where the total time elapsed includes, in addition to the time used in transmitting data traffic, the time incurred in collision and that used in transmitting control traffic. Another performance measure of interest is the connection establishment delay, t_r, defined as the time elapsed from the instant when a reservation request is made by a MH to the instant when the decision (on the corresponding slot assignment) is returned from the BS.

An effective MAC protocol should be designed so that all the transmissions take place with a bounded connection establishment delay, while only one transmission pair is active on the channel at any given time. The amount of control traffic generated and collision incurred should also be minimized in order to achieve high system utilization. Moreover, different levels of QoS should be supported.

2.2. Message Model

Let $\mathbf{M} = \{M_i \mid 1 \le i \le n\}$ be a set of real-time message streams to be scheduled by the BS. We use the (C, D)-smooth message model [5, 6] to characterize the traffic characteristics and the temporal QoS requirement of a message stream. In the (C, D)-smooth message model, each message stream M_i is characterized by, in addition to its source and destination nodes, a 2-tuple (C_i, D_i), where C_i is the maximum units of time needed for transmission of packets in M_i that may arrive in any time interval of length D_i, and D_i is the *relative* deadline for the packets in M_i, i.e., if a packet of M_i arrives at the MH at time t, then it must be delivered by time $t + D_i$.

This model is, in fact, a generalization of two commonly-used real-time traffic models: the *peak-rate* model, and the *linear bounded* model. The interested reader is referred to [5] for a detailed account, and a comparison, of the three message models. We also define the *message density* of an isochronous stream M_i as $\rho(M_i) = C_i/D_i$, and the total message density of a set of isochronous streams $\mathbf{M} = \{M_1, M_2, \ldots, M_n\}$ as

$$\rho(\mathbf{M}) = \sum_{i=1}^{n} \rho(M_i) = \sum_{i=1}^{n} \frac{C_i}{D_i}. \tag{1}$$

The following theorem proved in [5, 6] establishes the formal basis for scheduling transmission of a real-time message stream M_i with parameters (C_i, D_i) over a channel:

Theorem 1 *Suppose a message stream M_i conforms to the (C, D)-smooth model with parameters C_i and D_i. If at least C_i time units are allocated for transmission of packets in M_i in any time interval of length D_i, then each packet in M_i will be transmitted no later than D_i time units after its arrival.* □

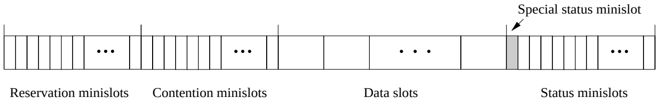


Figure 1. The frame structure used in QGMA.

3. Proposed Base Protocol

To present the key features of QGMA, we first present the base protocol with the following assumptions: (1) there exists only one frequency channel in each cell; (2) only the transmission from MHs to the BS is considered; (3) each MH requires only one class of service; and (4) the information on the flow/QoS specification is conveyed from a MH to the BS at the time when it registers with the BS. We will relax these assumptions in the next section.

There are other issues that should be considered in the design and implementation of wireless MAC protocols. For example, a forward error correction scheme or a local reliable retransmission scheme has to be incorporated in the protocol to guarantee reliable data delivery. Also, guard bits are usually required to be present between fields to ensure that a receiver can properly identify the fields from the byte stream it received. These issues are usually implementation dependent, and are beyond the scope of this paper.

3.1. Protocol Overview

In QGMA, an identification number, called *Host ID*, is used to uniquely identify each MH in a cell. When a MH enters a cell and registers itself with (and demands service from) the BS, it exchanges necessary information with the BS to set up the communication environment, e.g., the (universal) identity of the MH, the classes of service the BS can provide, the class of service the MH desires, and the *Host ID* the BS assigns to the MH. The *Host ID* is returned by a MH to the BS when it leaves the cell and/or is powered down. We denote the total number of *Host IDs* available in a cell as N_H and the MH who is assigned the *Host ID* n as MH_n.

The time on the frequency channel is divided into *frames*. As shown in Fig. 1, each frame is composed of four fields: reservation minislots, contention minislots, data slots, and status minislots. The first field of a frame is the N_R reservation minislots, each of which is of size L_R bits. The ith reservation minislot is exclusively used by mobile host MH_i to make its reservation request for a data slot on the channel. Since only one MH is eligible to send its reservation request in a reservation minislot, each request is guaranteed to reach the base station. On the other hand, since N_R may be less than N_H, some MHs may not have their dedicated reservation minislots. These MHs are labeled as *out-numbered* MHs, while those with *Host ID*s less than or equal to N_R are labeled as *in-numbered* MHs. Out-numbered MHs have to make their reservation requests on a contention basis in the contention minislots.

The second field of a frame is the N_D contention minislots, each of which is of size L_D bits. Contention minislots are used for out-numbered MHs to make reservation requests in a *slotted ALOHA* fashion. An out-numbered MH with a reservation request randomly chooses a contention minislot and sends its *Host ID*. If multiple MHs attempt to make reservation requests in the same contention minislot, a collision occurs, and the BS may or may not receive the reservation request.

The third field of a frame is the N_D data slots, each of which is of size L_D bits. Data slots are used to transmit data packets from MHs to the BS. The fourth field of a frame is the N_S status minislots, each of which is of

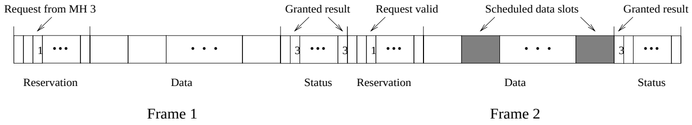


Figure 2. An example of the operations of QGMA.

size L_S bits. The status minislots are broadcast by the BS and are heard by all MHs. Each status minislot corresponds to a data slot in the *next* frame and contains the *Host ID* of the MH which is allowed to transmit its data packet in the corresponding data slot. Hence, the number, N_S, of status minislots in the current frame should be equal to the number of data slots in the next frame. The reservation minislots, the contention minislots, and the status minislots are the *system control overhead*.

3.2. Detailed Operations

If an in-numbered MH intends to transmit its data packet, it sends a reservation request in its reservation minislot. On the other hand, if an out-numbered mobile intends to transmit its data packet, it randomly chooses a contention minislot and sends its reservation request. After receiving all the (uncorrupted) requests in the reservation and contention minislots, the BS determines, based on the reservation requests received and the scheduling algorithm used, the slot schedule for the next frame. The resulting slot schedule will be broadcast in the status minislots of the current frame to the MHs which then begin their data transmission in the next frame accordingly.

An in-numbered MH has to send another reservation request if it would like to continue its transmission in the next frame. Upon completion of the transmission, an in-numbered MH simply resets its corresponding reservation minislot in the frame in which the host sends the last data packet. On the other hand, if an out-numbered MH succeeds in making reservation in one of the contention minislots, it will be henceforth granted data slots in all subsequent frames until the BS observes a data slot assigned to this MH is not used in a frame. At that instant, the BS assumes that the out-numbered mobile host has finished its transmission and will not assign any data packet to this host in subsequent frames. This avoids an out-numbered MH from suffering from an unbounded connection establishment for every data packet it sends. Once a request is received and processed by the BS, an out-numbered MH is guaranteed at its requested quality of service; no explicit requests need be made for subsequent data slots.

An example of how QGMA operates is demonstrated in Fig. 2. (For clarity of presentation, the contention minislots are not shown in Fig. 2.) MH_3 makes a reservation request in its reservation minislot in Frame 1. After receiving the reservation request, the BS decides to allocate two data slots to MH_3 in Frame 2 and informs MH_3 of the resulting schedule by putting MH_3's *Host ID* in the corresponding status minislots in Frame 1. MH_3 then starts to send its data packets in the data slots allocated in Frame 2. MH_3 also notifies the BS of that it would like to continue its transmission by making another reservation request in its reservation minislot in Frame 2. The BS thus continues to assign data slots to MH_3 and informs MH_3 of the resulting slot schedule in Frame 3 in the corresponding status minislot(s) in Frame 2.

3.3. Determination of Slot/Minislot Sizes

The identity of an in-numbered MH that makes a reservation request can be retrieved by a BS from the reservation minislot. Thus a reservation minislot needs only to carry one bit of information (to reserve or not to reserve), i.e., L_R can be as small as one. On the other hand, since an out-numbered MH must compete for a contention minislot to send its reservation request, it must include, in the minimum, its *Host ID* in the reservation request for the BS to identify the host. This means that L_C should be set to $\lceil log_2 N_H \rceil$, where N_H is the total number of *Host IDs* available in a cell.

Since each contention minislot occupies more bits than a reservation minislot and since MHs that compete for contention minislots may suffer from unbounded connection establishment delays, it may be desirable to eliminate the contention minislots totally. However, the number of MHs present in a cell must then be less than or equal to N_R, which may not be true in a dynamically changing environment. We will propose a dynamic frame structure in Section 4 in which the number of reservation minislots is dynamically adjusted based on the current number of MHs in a cell.

L_D may be set to the size of packets (or cells) carried in the point-to-point backbone network, while the decision on N_D is a tradeoff between the system utilization (where a large value of N_D is desired) and the connection establishment delay (where a small value of N_D is desired). Since the kth ($1 \leq k \leq N_S$) status minislot carries the *Host ID* of a MH that is granted the kth data slot in the next frame, L_S should be set to $\lceil log_2 N_H \rceil$ and N_S should be equal to N_D.

3.4. Features of QGMA

QGMA possesses several desirable features. First, a reservation request for establishing a time-critical message stream from a MH must be received (and processed) by the BS within a bounded time interval. In QGMA, a reservation request made in a reservation minislot is guaranteed to be received by the BS, and the corresponding assignment will be returned to the MH in the status minislots in the same frame. That is, the connection establishment delay t_r is less than a frame time. A reservation request made in a contention minislot has to compete against others and may, as in all the existing MAC protocols, suffer from an unbounded connection establishment delay. The problem can be eliminated by having the BS maintain a handful of in-numbered *Host IDs* and only assigning to MHs with real-time requirements in-numbered *Host IDs*. (The number of *Host IDs* may be determined based on the bandwidth available to handle real-time applications.) An alternative approach is to dynamically adjust the frame structure to accommodate all MHs as in-numbered hosts, the detail of which will be discussed in Section 4.1.

Second, sufficient bandwidth (data slots) must be provided to a MH with a real-time message stream to guarantee the timely delivery of its data packets. QGMA achieves this by having the BS take charge of data slot assignment and scheduling. The QoS required by a MH is conveyed (in terms of a set of traffic/QoS parameters) to the BS when the MH registers itself with the BS. Serving as a centralized data slot scheduler and having the overall control of all the channel resources, the BS then determines, based on a specific scheduling algorithm, the appropriate number of data slots to be allocated to each MH. (Multiple data slots may be allocated to a MH in a frame.) Different levels of QoS can be supported by utilizing different scheduling algorithms. For example, rate-based flow control can be achieved by having the BS use the *packet-by-packet generalized processing sharing* (PGPS) scheduler. Applications with *relative deadlines* can be easily supported by having the BS employ the *distance-constrained task scheduling* (DCTS) algorithm [6]. We will demonstrate how to incorporate a DCTS-based scheduler, called **SlotAllocator**, into QGMA in Section 5. The fact that QGMA is *flexible* to incorporate different scheduling algorithms for allocating data slots makes it well suited for a wide spectrum of applications that require temporal QoS.

4. Extensions on the Base Protocol

In this section, we discuss (i) a dynamic frame adjustment scheme to support timely establishment of message streams (Section 4.1), (ii) the issue on how to convey different levels of QoS required by a MH to the BS (Section 4.2), (iii) the issue on how to incorporate the MH registration process into QGMA (Section 4.3), (iv) the issue on how to support traffic from the BS to MHs (Section 4.4), (v) the issue on how to support MHs with multiple message streams, and (vi) the issue on how to support the case of separate upstream and downstream channels. Due to space limit, we present in this paper only extension (i)–(iv). A complete discussion on the extensions can be found in the full version of the paper [12].

4.1. Support for Timely Establishment of Message Streams

As was discussed in Section 3, since reservation requests made in contention minislots may suffer from unbounded connection establishment delays and a contention minislot requires more bits than a reservation minislot, it may be desirable to eliminate contention minislots in order to increase system utilization and to support real-time applications. However, a MH that arrives at a crowded cell in which all the N_R reservation minislots are allocated/used may be unnecessarily blocked in this case even if the bandwidth available on the channel is sufficient.

To solve this problem, we propose to dynamically adjust the frame structure. If the number of active MHs in a cell is more than the reservation minislots available at certain point, the protocol simply announces a new frame structure with enough reservation minislots and uses it in the next frame. Every MH can then be offered a dedicated reservation minislot. On the other hand, when the BS detects that the number of active MHs is far less than that of the valid *Host IDs*, it announces a new frame structure with less reservation minislots. The announcement is made by inserting before the status minislots in a frame an *special* status minislot that gives the information on the field sizes in the next frame (e.g., the number of reservation minislots, N_R, the number of contention slots, N_C, and the number of data slots, N_D). After the announcement is made, all the MHs in the cell will be able to tell the boundaries of status minislots, reservation minislots and/or data slots in the next frame.

Note that since the number of status minislots in the current frame equals the number of data slots in the next frame, MHs may not be able to determine the number of status minislots in the current frame until they know exactly the number of data slots in the next frame. This is the reason why the special minislot should be placed before the status minislots. As will be demonstrated in Section 5, dynamic adjustment on the frame structure is especially well-suited for the case in which BSs incorporate different scheduling algorithms to support different levels of QoS.

4.2. Support for Multiple Classes of Services

We propose two approaches to supporting multiple classes of services. In the case that only a set of n explicit classes of services exist in the WLAN, the BS may provide the list of n classes available when a MH registers with it. Each reservation minislot in the frame is now made $\lceil \log_2 n \rceil$ bits long. Whenever a MH makes a request reservation, it includes the class of service desired in the reservation minislot. Multiple classes of service are supported at the expense of reservation minislot overheads.

In the case that the quality of service (e.g., delay bounds and delay jitter bounds) cannot be completely characterized only by a set of quantized service classes, a MH must inform the BS of the QoS required in terms of a set of parameters in its reservation request. The proposed base protocol can be extended to accommodate this case by using a two-step connection establishment approach: a MH first makes a reservation request in frame F and indicates that a special data packet that contains the QoS parameters will be sent to the BS in the next data slot assigned to this MH. This can be done by defining a special (*flow specification*) request type in reservation minislots for the use of passing QoS parameters. If a data slot is allocated to the MH in frame $F + 1$, the host will then send the QoS parameters to the BS which then determines, based on certain scheduling algorithm, the slot schedule and informs the MH of the resulting slot schedule in the status minislots in frame $F + 1$. The MH can then begin transmission of its message stream in frame $F + 2$.

The above approach is an example of *in-band signaling* in which the signaling information (i.e., the QoS parameters) is passed from a MH to the BS using the traffic bandwidth (data slots) instead of the signaling bandwidth (reservation and/or status minislots). Using the in-band signaling approach, a MH can send a large amount of control information to the BS whenever needed without defining and incurring fixed control overheads in all the frames.

In the case that a MH needs to change the QoS parameters after a connection is established, the MH simply sends a new flow specification request in the reservation minislot followed by the data packet that contains the new QoS parameters. The BS will then update the data slot schedule accordingly to adapt to the changing QoS demand.

4.3. Support for MH Registration

As discussed in Section 3, when a MH first enters a cell, it has to register itself with the BS to obtain its *Host ID* and to convey its traffic/QoS characteristics. This registration process can be easily realized in QGMA through the use of contention minislots and in-band signaling (discussed in Section 4.2).

A MH initiates the registration process by sending its global unique identification (GUID) over a contention minislot it randomly chooses. Upon receiving the GUID from a MH, the BS assigns the MH a *Host ID*, and sends it, along with the other information (e.g., classes of services provided) in a data slot to the MH. Note that the BS uses, instead of the assigned *Host ID*, the GUID as the identification in the data slot. After a MH successfully sends its GUID in a contention minislot, it continues to monitor the channel until it receives a data packet destined to it. The MH then extracts, among other relevant information, the *Host ID* from the data packet and completes the registration process. (The MH can proceed to convey its QoS requirement using the methods proposed in Section 4.2.)

In the dynamic frame structure, a few contention minislots can be designated and used exclusively for the purpose of MH registration, while all the reservation minislots are used for reservation. Due to the fact that usually only a few new MHs arrive in a frame time, a number of 2-4 contention minislots per frame is sufficient to ensure a reasonably small registration delay.

4.4 Support for Traffic from the BS to MHs

Since the BS is responsible for generating the slot schedule, it knows in which data slots it should transmit to a MH. The only problem is how a MH know in which data slots it is supposed to receive data from the BS.

There are two possible solutions: in the first solution, the BS assigns a unique *Host ID* to itself and makes this *Host ID* known to all the MHs. When the BS intends to transmit data packets to a MH, it simply puts its *Host ID* in the corresponding status minislots. Also, the *Host ID* of the destination MH is included in the headers of data slots. After a MH observes the *Host ID* of the BS in a status minislot, it listens to the corresponding data slot to see if the data packet is destined to it. This method does not require any change in the base protocol, but requires that all the MHs monitor the traffic originating from the BS.

The second solution requires a little modification to the base protocol. Each status minislot is modified to contain a flag, indicating the direction of transmission that will take place in the corresponding data slot in the next frame. If the BS intends to send a data packet to MH MH_i, it simply sets the flag in the corresponding status minislot to 1 and puts the *Host ID* (i) of the destination MH in the status minislot. After MH_i observes the status minislots, it knows which data slots will be used for traffic from the base station to MH_i. Only MH_i (and no other MHs) will monitor those data slots for data reception. This method, however, adds a flag bit to each status minislot and increases the system control overhead.

5. Incorporation of a DCTS-Based Scheduler to Support Temporal QoS

In this section, we first summarize the distance-constrained task system (DCTS) model used to characterize tasks with temporal constraints and the corresponding DCTS scheduling algorithm, and then demonstrate how to incorporate DCTS into QGMA to support temporal QoS.

For clarity of presentation on the key concept of DCTS, we assume that all the reservation requests as well as the flow/QoS specifications for the set of real-time message streams under consideration have been passed to the BS using the methods discussed in Section 4.2. To deal with dynamic changes in a mobile environment and to ensure that all the reservation requests reach the BS in a timely fashion, we use the dynamic frame structure discussed in Section 4.1. Under the above assumptions, we then concentrate on how the base station uses the DCTS-based scheduler for slot assignment.

5.1. The DCTS-Based Scheduler

We devised in [5] an DCTS-based on-line slot allocator, called **SlotAllocator**, which can generate, for a given set of message streams $\mathbf{M} = \{M_i = (C_i, D_i) \mid 1 \le i \le n\}$, a slot allocation schedule that satisfies the criterion that for any consecutive D_i slots, exactly C_i slots are allocated to M_i, for all i as long as D_i divides D_j for all $i < j$ and $\rho(\mathbf{M}) \le 1$. (The constraint on D_i divides D_j for all $i < j$ will be relaxed below.)

Succinctly, **SlotAllocator** uses the well-known *rate-monotonic* scheduling algorithm [7] and treats C_i as the computation time and D_i as the period of a task. It assigns priorities to message streams so that the streams with tighter deadlines get higher priorities, i.e., if $D_i < D_j$ then M_i has a higher priority than M_j (ties are broken arbitrarily). After the system is initialized, **SlotAllocator** assigns C_i slots to a message stream M_i during each time period $[(j-1)\cdot D_i, j\cdot D_i]$, for all $1 \le i \le n$ and all $j \ge 1$. This is done by assigning the current slot, say $[t-1, t]$, to the message stream with the highest priority among all the *active* message streams, where an active stream M_i is one whose slot requirement with respect to its current time period is *unfulfilled*, i.e., from time $(j-1)\cdot D_i$ to time $t-1$, there are less than C_i slots assigned to M_i, where $(j-1)\cdot D_i \le t-1 < j \cdot D_i$ for some integer j. We showed in [5] that **SlotAllocator** has a time complexity of $O(n)$ per slot generated, and establish the theoretical basis for **SlotAllocator** in the following theorem:

Theorem 2 *For a set of real-time message streams* $\mathbf{M} = \{M_i = (C_i, D_i) \mid 1 \le i \le n\}$, *if* D_i *divides* D_j *for all* $i < j$ *and* $\rho(\mathbf{M}) \le 1$, **SlotAllocator** *will allocate* C_i *slots to* M_i *in any time window of size* D_i *slots, for all* i. □

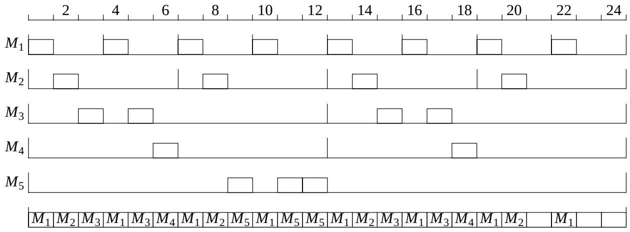

2
4
6
8
10
12
14
16
18
20
22
24
M_1
M_2
M_3
M_4
M_5
M_1 M_2 M_3 M_1 M_3 M_4 M_1 M_2 M_5 M_1 M_5 M_5 M_1 M_2 M_3 M_1 M_3 M_4 M_1 M_2 M_1

Figure 3. The slot allocation schedule for Example 1.

For an arbitrary set of real-time message streams, **M'** $= \{M'_i = (C_i, D'_i) \mid 1 \le i \le n\}$, in which the deadline constraint set **D'** $= \{D'_1, D'_2, \ldots, D'_n\}$ (without loss of generality, we assume $D'_i \le D'_j$ for all $i < j$) does not necessarily consist solely of multiples (i.e., D'_i divides D'_j may not be true for all $i < j$), we first transform the arbitrary message set **M'** to another message set $\mathbf{M} = \{M_i = (C_i, D_i) \mid 1 \le i \le n\}$, in which the transformed deadline constraint set $\mathbf{D} = \{D_1, D_2, \ldots, D_n\}$ consists solely of multiples and $D_i \le D'_i$ for all i. The operation, termed as *specializing* **D'** (**M'**) *with respect to* $\{x\}$ [5], finds a D_i for each D'_i such that D_i satisfies $D_i = x \cdot 2^j \le D'_i < x \cdot 2^{j+1} = 2D_i$, for some integer $j \ge 0$, where x is an integer $\in (D'_1/2, D'_1]$ that results in the minimum total density increase.[1] Since **D** is more restricted than **D'**, if we find a feasible slot allocation schedule for **M**, then the schedule is also feasible for the original constraint set **D'**. We illustrate the operations of **SlotAllocator** with the following example.

Example 1 *Consider a set of real-time message streams,* **M'** $= \{$*(1,4), (1,7), (2,13), (1,23), (3,28)*$\}$.[2] *We first specialize the deadline constraint set* **D'** $= \{$*4, 7, 13, 23, 28*$\}$ *with respect to* $\{3\}$ *to* **D** $= \{$*3, 6, 12, 12, 24*$\}$. *Since* D_i *divides* D_j, *for all* $i < j$, *and* $\rho(\mathbf{M}) = \sum_{i=1}^{5} C_i/D_i = 1/3 + 1/6 + 2/12 + 1/12 + 3/24 = 21/24 < 1$, *by Theorem 2, we know that* **SlotAllocator** *can find a feasible schedule for* **M**. *Using* **SlotAllocator**, *we obtain the slot allocation schedule as shown in Fig. 3 in which the schedule repeats every* $D_5 = 24$ *slots. Note that* **SlotAllocator** *always assigns a slot to the message stream with the tightest deadline among all active message streams. As one can readily see, there are at least* C_i *slots assigned to* M_i *in any time window of size* D_i $(\le D'_i)$ *slots, and hence, in any time window of size* D'_i *slots.* □

SlotAllocator, coupled with the specialization operation, is simple, effective, and provides a simple schedulability check: as long as the total message density $\rho(\mathbf{M})$ after specialization is less than or equal to 1, the deadline constraints for all streams can be guaranteed. In what follows, we discuss how to incorporate the DCTS-based **SlotAllocator** into QGMA.

5.2. Incorporation of the DCTS-Based Scheduler

When **SlotAllocator** generates the slot schedule, it assumes that all the time slots are available for data transmission. However, certain system control overhead (reservation/contention/status minislots) is incurred for transmitting signaling information in QGMA. One problem that arises in order to incorporate **SlotAllocator** is thus how **SlotAllocator** takes into

[1] Details on how to determine the value of x can be found in [5].

[2] We omit the source and destination stations, N_i^s and N_i^d, for each M_i since they are irrelevant in this example.

account of the system control overhead when it generates a slot schedule. In addition, we need to determine the frame size and the size of system control overhead in each frame.

The system control overhead is composed of two continuous blocks in each frame. The first block consists of reservation minislots (and perhaps a few contention minislots as discussed in Section 4.3), and the second block consists of status minislots. To include the system control overhead into the slot schedule, we may visualize the status minislots in frame F and the reservation minislots in frame $F+1$ as one indivisible continuous block. We then define a (virtual) system message stream M_0 with parameters (C_0,D_0), where D_0 is the frame size and C_0 is the size of system control overhead in each frame, both of which are yet to be determined. We also impose an upper bound D_H on the frame size D_0 in order to reduce the connection establishment delay.

Although determination of C_0 and D_0 may depend on the message set $\mathbf{M}$, the *overhead ratio* C_0/D_0 can be approximated as a constant, because

$$\begin{aligned}\frac{C_0}{D_0} &= \frac{L_R \cdot N_R + L_S \cdot N_S}{L_R \cdot N_R + L_S \cdot N_S + L_D \cdot N_D} \\ &\overset{L_R = 1,\, N_S = N_D}{=} \frac{N_R + L_S \cdot N_D}{N_R + L_S \cdot N_D + L_D \cdot N_D} \\ &\approx \frac{L_S}{L_S + L_D}.\end{aligned}$$

Note that the above approximation results from the fact that N_R is usually negligible as compared with N_D. As both the values of L_S and L_D are fixed, so is C_0/D_0.

The schedulability check can now be modified as follows. Given an arbitrary message set $\mathbf{M}' = \{M_i' = (C_i, D_i') \mid 1 \le i \le n\}$, we first specialize $\mathbf{M}'$ to another message set $\mathbf{M} = \{M_i = (C_i, D_i) \mid 1 \le i \le n\}$ in which $D_i \mid D_j$ for all $i < j$. Then, we check whether or not

$$\rho(\mathbf{M}) \le 1 - \frac{C_0}{D_0} = 1 - \frac{L_S}{L_S + L_D}.$$

If the above criterion is met, the set of message streams can be established.

Now the problem is to determine the frame size D_0 and the system overhead size C_0. Specifically, given a set of real-time message streams $\mathbf{M} = \{M_i = (C_i, D_i) \mid 1 \le i \le n\}$ and a system message stream M_0 with a fixed overhead ratio C_0/D_0 (both of which are to be scheduled by **SlotAllocator**), we have to determine D_0 (and hence C_0) such that

(1) The slots allocated to M_0 constitutes an indivisible continuous block of size C_0 in each frame of length D_0.

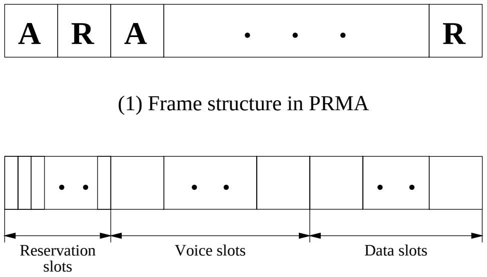
(2) $D_0 \le D_H$ and yet is as large as possible to allow enough time for the BS to generate the slot schedule.

To fulfill the first requirement, we must set M_0 at the highest priority among all the other real-time message streams so that **SlotAllocator** will allocate consecutive slots to M_0. Since **SlotAllocator** uses the rate monotonic scheduling algorithm, the requirement that M_0 be assigned the highest priority implies that $D_0 < D_i, i = 1, \cdots, n$. Moreover, to conform to the property that the specialized message set consists solely of multiples, we must set $D_0 = D_1/m$, where $m \ge 1$ is a positive integer. An expression that satisfies both (1) and (2) is $D_0 = \frac{D_1}{\lceil D_1/D_H \rceil}$. The first requirement is fulfilled because

$$\begin{aligned} m = \lceil \frac{D_1}{D_H} \rceil \ge 1 \quad &\Rightarrow \quad D_0 = \frac{D_1}{\lceil D_1/D_H \rceil} \le D_1 \\ &\Rightarrow \quad D_0 \le D_i, i = 1, \cdots, n. \end{aligned}$$

The second requirement is met because

$$\lceil \frac{D_1}{D_H} \rceil \ge \frac{D_1}{D_H} \quad \Rightarrow \quad D_0 = \frac{D_1}{\lceil D_1/D_H \rceil} \le \frac{D_1}{D_1/D_H} = D_H,$$



(1) Frame structure in PRMA

(2) Frame structure in D-TDMA

Figure 4. Frame structures for PRMA and D-TDMA.

and also that $m = \lceil D_1/D_H \rceil$ is the smallest positive integer that satisfies $D_1/m \le D_H$. To see this, we need to show that $D_1/(m-1) > D_H$:

$$m = \lceil \frac{D_1}{D_H} \rceil < \frac{D_1}{D_H} + 1 \quad \Rightarrow \quad \frac{D_1}{m-1} > D_H.$$

After both the values of D_0 and C_0 are determined, the BS then applies the **SlotAllocator** to generate a slot schedule for the message stream set, $\mathbf{M} \cup \{M_0\}$. As long as the message stream set $\mathbf{M}$ stays unchanged, the frame structure remains unchanged. Whenever a message stream is to be established/terminated, the BS will re-compute the frame structure and generates a new slot schedule for the new message stream set.

6. Survey on Existing MAC Protocols

Several MAC protocols have been proposed in wireless networks, among which Packet Reservation Multiple Access (PRMA) [8], Dynamic TDMA (D-TDMA) [11], Dynamic Reservation Multiple Access (DRMA) [10], Resource Auction Multiple Access (RAMA) [1], Floor Acquisition Multiple Access (FAMA) [3], and Remote-Queuing Multiple Access (RQMA) [4] may receive the most attention. We provide a brief overview on them, and compare our proposed protocol against them.

PRMA: The first MAC protocol proposed to support wireless communication is the PRMA protocol [8]. As shown in Fig. 4 (1), in PRMA, time is divided into *slots* and several slots form a *frame*. Users are classified into two types: voice users and data users. Each user with voice/data to transmit simply randomly chooses a slot available inside a frame for packet transmission. Whether or not contention occurs in a slot will be known to all the senders by the end of the slot. If a voice user successfully transmits its voice packet in a slot, the slots will be labeled as reserved in subsequent frames until released by the voice user upon completion of its transmission. However, this rule does not apply to data users, who are required to contend for every slot it would like to use for data transmission. Due to its CSMA nature, PRMA suffers from low utilization in medium to heavy traffic loads and does not provide a deterministic bound on the connection establishment delay.

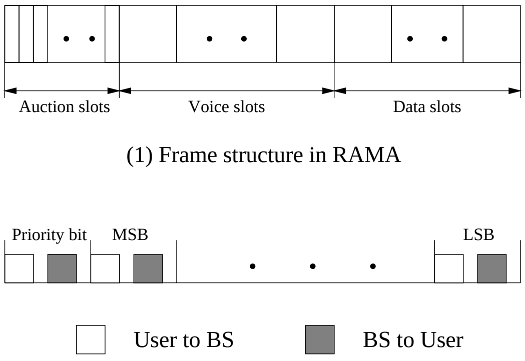
D-TDMA: As shown in Fig. 4 (2), in D-TDMA, time is divided into frames, and each frame is composed of reservation slots, voice slots and data slots. To reserve an information (voice/data) slot, a user sends in a randomly chosen reservation slot a reservation packet. The reservation packet contains information needed to establish a connection, e.g., the source/destination addresses. At the end of a reservation period, successful reservation will be identified and the final slot schedule will be broadcast to all the users by the BS. Due to the slotted ALOHA nature, the successful rate for making reservation deteriorates in the case of medium to heavy traffic loads. Moreover, successful transmission of a reservation packet does not guarantee that a user be always allocated an information slot due to the limited number of information slots available. Once allocated a voice slot, a user can use the same slots in subsequent frames until it completes its transmission, while a data user is granted one data slot (in the same frame as it makes the reservation) at a time. Unsuccessful users will retry in the next frame according to the *reservation retransmission* probability.

RAMA: RAMA is very similar to D-TDMA except that it uses a different reservation approach. As shown in Fig. 5 (1), reservation slots in a frame are replaced by auction slots in RAMA. In each auction slot, the available resources (i.e., information slots) will be auctioned to requesting users and will be assigned to the winner of the auction. The auction procedure works as follows (Fig. 5 (2)): each requesting user is assigned a user ID which is randomly generated when the user decides to attend the auction. The number of digits used in the random number depends on the number of users currently in the network. Requesting users start to transmit their IDs in the auction slots, one at a time, from the most significant bit to the least significant bit. After each bit transmission, the BS broadcasts the largest bit value it receives just now, and those contending MHs with unmatched bit value will drop off. The final winner at the end of each auction slot will not attend any future auction in the same frame. The users dropping off in the current auction slot can select another random number and reenter the auction in the next slot.

One attractive property of RAMA is that one MH will finally win out in each auction and be successful in sending its reservation request to the BS. However, due to the randomness nature of the auction process, there is no deterministic bound on the time needed for a requesting user to finally win the auction.

DRMA and FAMA: DRMA is a variation of the above protocols, and differs in the degree of design complexity and the level of bandwidth efficiency thus achieved. DRMA eliminates the reservation/auction slots in D-TDMA/RAMA, and uses (if necessary) an available slot as a set of reservation slots. Efficiency is achieved by dynamically assigning reservation slots, rather than using fixed reservation slots. FAMA, on the other hand, basically applies the carrier sense multiple access with collision detection mechanism to the control and jamming packets sent from MHs to the BS, and can be regarded as a CSMA/CD scheme in a WLAN.

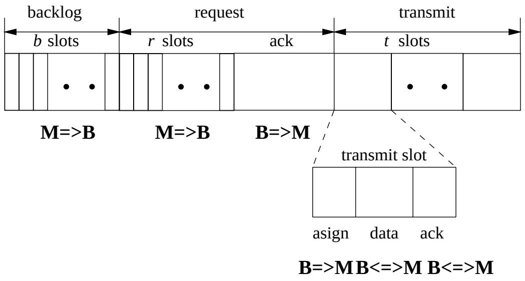
All the above wireless MAC protocols are tailored to meet the specific requirement of supporting only voice and data users, and do not address the need for supporting other aspects of QoS. In particular, they provide neither deterministic bounds on the time by which a reservation request can reach the base station nor temporal QoS required by embedded or multimedia-integrated real-time applications. Recently, several other MAC protocols have been proposed that address the QoS issues in wireless LANs, which we summarize below.



(1) Frame structure in RAMA

(2) One auction slot

Figure 5. The frame structure in RAMA.



Frame structure in RQMA

Figure 6. The frame structure in RQMA

RQMA: RQMA, with the design objective of supporting QoS services, shares several common features with QGMA. In particular, both protocols support multiple levels of QoS services by incorporating real-time scheduling algorithms in slot assignment. RQMA supports three types of traffic: constant bit rate (CBR), real-time, and best-effort. A frame in RQMA is divided into three fields: b backlog slots, r request slots (and their corresponding ack subfields), and t transmission slots (Fig. 6). A MH sends a request in a request slot (in a slotted ALOHA fashion) to the BS to either establish a RT/CBR session or send best-effort packets. If the BS successfully receives a request in a request slot, it sends an ack in the corresponding ack subfield. In the case that a real-time session is established, a MH uses one backlog slot (assigned by the BS) to inform the BS of any newly-arrived packets of the real-time session and their deadlines. The BS then determines when a MH can send/receive data packets of a session, by specifying in the *assign* subfield of each transmit slot the MH id and the session id.

The most notable feature of RQMA [4] is that it takes into consideration of the error characteristic of the wireless channel, and establishes *a priori* a *real-time retransmission* session to retransmit time-critical data packets upon error detection in the normal transmission phase. There are, however, several drawbacks in RQMA: first, to support temporal QoS, MHs are required to calculate packet deadlines by themselves. This may not be desirable because MHs are usually small and inexpensive devices and may not possess much power to perform deadline calculation. Second, in order to accommodate error control, RQMA has a rather inefficient frame structure as compared to QGMA. It can be calculated from Table 1 (which lists the contents of each field in a frame) in [4] that the maximum achievable utilization in RQMA is approximately 58% (i.e., 424 data bits/733 bits in a frame, excluding the FEC bits to make the comparison fair) while it can be as high as 90% in QGMA (Fig. 7). Third, RQMA uses contention slots in a slotted ALOHA fashion for sending session setup requests, and thus suffers from unbounded connection establishment delay.

7. Performance Evaluation

QGMA is evaluated according to the following sequence: (1) We first compare the performance between QGMA and the other MAC protocols (PRMA and RAMA, in particular) under voice traffic. (2) We demonstrate the capability of supporting temporal QoS (in terms of packet loss ratio) when QGMA is coupled with DCTS, rate monotonic (RM), and round-robin (RR), respectively.

As the first MAC protocol designed for wireless LAN, PRMA is simple and yet quite satisfactory in handling voice traffic on wireless network. RAMA, on the other hand, is reported to achieve, in general, the best performance among all the existing MAC protocols. The performance gain of RAMA partially results from the fact that RAMA assumes that with support of specialized hardware, one contender can still win out in the case of channel collision and has its message delivered successfully. (This is termed in [10] as the *collision recovery property*). RAMA reduces to D-TDMA with this feature turned off. DRMA has similar performance to RAMA [10]. In the case that the reservation overhead is a significant portion of the total channel bandwidth, DRMA performs slightly better than RAMA due to its flexibility in adjusting reservation and data bandwidth [10]. However, as the reservation overhead usually occupies only a negligible portion of the total channel bandwidth, it suffices to choose RAMA for comparison. (As discussed in Section 6, RQMA suffers from low utilization, and is thus excluded from the comparison.)

In the event-driven simulation, MHs arrive at a cell according to a Poisson process with rate λ. The interval in which a MH stays in the cell is exponentially distributed with rate μ. (The MH arrival process can then be modeled as a $M/M/\infty$ queue. In steady state, the number of MHs in a cell is Poisson distributed with parameter λ/μ.) We assume that the channel bandwidth available for both data and signaling in a cell is 600 Kbps. We also assume that no error occurs in packet transmission. Packet corruption is only caused by collisions.

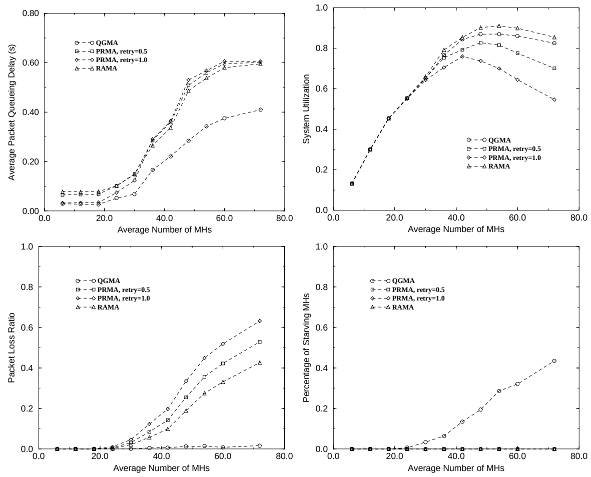


Figure 7. Performance comparison between QGMA, PRMA, and RAMA. $\gamma_t = \gamma_s$.

The traffic used in the first set of experiments is voice traffic from MHs to the BS. The voice terminal model [8] is used to model voice traffic in which a voice user generates talkspurt and silence gap alternately. We assume that the lengths of talkspurt and silence gaps are both exponentially distributed with rate γ_t and γ_s, respectively. When the user is in a talkspurt, it generates a constant rate voice traffic of 30 Kbps; otherwise (during a silence gap), it does not generate any traffic. As a result, the effective bandwidth required by a voice user is $30 \cdot \frac{\gamma_s}{\gamma_t+\gamma_s}$ Kbps. A voice packet generated by a user is considered lost if it cannot be transmitted in 1 second. The traffic used in the second set of experiment is the (C, D)-smooth traffic (Section 2.2) from MHs to the BS. The ratio of the parameters C and D is an exponentially distributed random variable with rate δ ($\delta = C/D$) and is denoted as the *density* of the (C, D) message stream.

In the first simulation run, we collect statistics and compute (i) the average packet queuing delay, defined as the time elapsed between the arrival of a voice packet and its transmission (lost packets are not counted in this measure), (ii) the average system utilization; (iii) the packet loss ratio; and (iv) the percentage of starving MHs, where a MH is said to be *starved* if its connection requests, although received by the BS, are not approved by the BS due to limited channel bandwidth. In the second simulation run, we apply the scheduling algorithms DCTS, RM and round-robin to QGMA and collect statistics of (i) the packet loss ratio; (ii) the percentage of MHs who suffer from packet loss; and (iii) the percentage of MHs who failed the admission test and were rejected (which is denoted as *MH failure rate*).

7.1. Comparison Among QGMA, PRMA, and RAMA

We simulate two instances of PRMA with retry probability of 0.5 and 1.0, respectively. Each frame in PRMA contains 20 data slots (with each slot set to 1,000 bits). Each frame in RAMA contains 20 data slots (with each slot set to 1,000 bits) and 40 reservation minislots (with each reservation minislot set to 10 bits). In QGMA, the parameters are assigned as follows: (i) a reservation minislot is set to 1 bit long and the number of reservation minislots (N_R) in a frame is set to the maximum number of MHs that the cell can accommodate with the bandwidth available, i.e., $N_R = N_H = 600/(30 \cdot \frac{\gamma_s}{\gamma_t+\gamma_s})$; (ii) a contention minislot is set to 10 bits long and the number of contention minislots (N_C) in a frame is (arbitrarily) set to N_R; and (iii) a status minislot is set to $N_S = \log_2 N_H$, and the number of status minislots (N_S) in a frame equals N_D which is arbitrarily set to N_H. Both RAMA and QGMA use the round-robin algorithm for scheduling voice packets. Note that all the above parameters are chosen, whenever possible, to tune the specific protocol to its best performance, given the voice traffic considered.

Fig. 7 gives the performance comparison between QGMA, PRMA, and RAMA in terms of average packet queuing delay, system utilization, packet loss ratio, and ratio of starving mobile hosts for a wide range of traffic loads. Each data point in the figures is obtained by generating 2,500,000 voice packets in the simulation run. The effective voice bandwidth per user is $30 \cdot \frac{\gamma_s}{\gamma_t+\gamma_s} = 30/2 = 15$ Kbps. It can be seen from both figures that QGMA significantly outperforms the others in terms of packet queuing delay and packet loss ratio. This is because QGMA provides dedicated reservation minislots for in-numbered MHs to make reservation requests, thus achieving a bounded connection establishment delay and a low packet queuing delay. RAMA has the best system utilization, while QGMA is only slightly inferior to RAMA. The reason that RAMA performs best is perhaps due to its collision recovery property, i.e., with support of specialized hardware, RAMA assumes that one contender can still win out in the case of channel collision and has its message delivered successfully. QGMA achieves high system utilization because it eliminates channel collision by providing dedicated reservation minislots to MHs.

On the other hand, QGMA has the highest ratio of starving MHs. This is because in order to provide QoS guarantees for existing mobile voice users in a cell, QGMA does not grant slot access to new MHs (which will observe long connection establishment delays) in the case that the current number of MHs in the cell is greater than or equal to N_H. (The QoS achieved at the expense of starving new MHs is evidenced in the significantly lower packet loss ratio.) Due to the random access nature, PRMA and RAMA do not prevent new MHs from competing with existing MHs for channel bandwidth and hence their ratio of starving MHs is always zero. (However, also due to the same reason, PRMA and RAMA cannot provide any QoS guarantees to mobile voice users.) QGMA can be "tuned" to accommodate more mobile voice users at the expense of compromising the QoS for each mobile voice user, should that be necessary.

7.2. Comparison Between DCTS, RM and RR

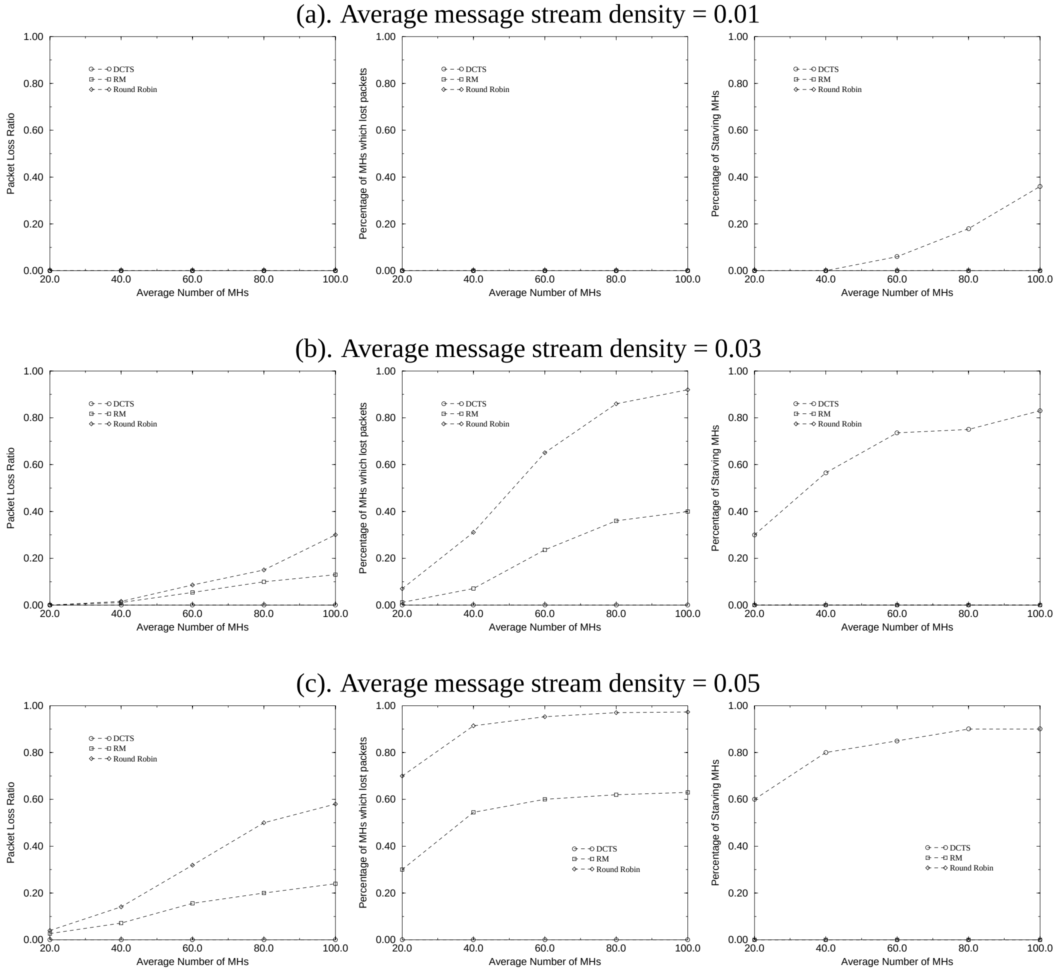
In this simulation run, we compare the capability of RM, round-robin (RR), and DCTS in scheduling the (C, D) traffic when they are incorporated into QGMA. The RR scheduler is a simple yet effective scheduler in scheduling constant rate periodic traffic, e.g., voice traffic. However, by giving equal priority to any message stream, RR does not perform well when it deals with real-time traffic. RM is one of the earliest proposed scheduling algorithms that are capable of supporting real-time traffic. It has been proved in [7] that as long as the total message density of a set of n message streams is less than or equal to $n(2^{1/n} - 1)$, RM guarantees that no packet loss due to missing deadlines may occur. However, this schedulability test is only a sufficient condition but not a necessary condition. (That is, if the total density of a message stream set exceeds $n(2^{1/n} - 1)$, no conduction can be drawn on whether or not the message stream set is schedulable by RM.) DCTS, on the other hand, extends the schedulability test further: if a set of (C, D) message streams satisfies the above schedulability test, it is schedulable by DCTS. Moreover, if the set of message streams fails the schedulability test but after specialization has a total density ≤ 1, the message stream set is still schedulable by DCTS.

In this simulation run, we apply the generated traffic to each of the three scheduling algorithms and collect statistics of (i) the packet loss ratio; (ii) the percentage of MHs who experience packet loss (due to missing deadlines) during their stay in the system; and (iii) the percentage of MHs that are rejected by the schedulability test of DCTS (which is denoted as *MH failure rate*). Note that RM and RR does not reject any MHs since they do not perform schedulability tests. We perform three simulation runs in which the average message density ($\delta = C/D$) is set to be 0.01 (low load), 0.03 (medium load) and 0.05 (high load), respectively.

The performance comparison between DCTS, RM and RR is presented in Fig. 8. Each data point is obtained by simulating 1,000,000 packets. (Note that in the DCTS case, only MHs that pass the admission test are included in calculating packet loss.) It can be observed that with the help of the schedulability test, DCTS provides guaranteed QoS to the message streams it schedules (packet loss ratio is 0). The cost of this guaranteed QoS comes from the high rejection rate of MHs. Without blocking any MHs, RM and RR suffer from high packet loss ratio as well as high MH failure rate under medium to high traffic loads. RM performs better than RR (i.e., the packet loss ratio and the MH failure rate under RM are smaller than those under RR) since it prioritizes message streams based on message deadlines.

8. Conclusions

In this paper, we propose a novel MAC protocol, called QGMA, in wireless local area networks to support the quality of service required by embedded and multimedia-integrated real-time applications. As compared to other existing MAC protocols, QGMA achieves higher system utilization, and provides a deterministic bound on the connection establishment delay and temporal QoS for real-time applications. We validate our assertions through event-driven simulations. QGMA is also flexible in the sense that it encompasses several existing MAC protocols. For example, in the case that no reservation minislots exist and the


(a). Average message stream density = 0.01

(b). Average message stream density = 0.03

(c). Average message stream density = 0.05

Figure 8. Performance comparison between DCTS, RM and RR.

BS generates the slot schedule on a first-come-first-served basis, QGMA reduces to D-TDMA. In the case that no reservation minislots exist and the auction scheme is applied to the contention minislots, QGMA reduces to RAMA.

To demonstrate how to incorporate a scheduling algorithm into QGMA, we use the distance constrained task scheduling algorithm as an example and discuss in a step-by-step manner how to incorporate it into QGMA. We are currently investigating a general methodology for incorporating various scheduling algorithms into QGMA.

References

[1] N. Amitay. Distributed Switching and Control with Fast Resource Assignment/Handoff for Personal Communications Systems. *IEEE JSAC*, 11:842–849, 1993.

[2] M. Y. Chan and F. Chin. Schedulers for Larger Classes of Pinwheel Instances. *Algorithmica*, 9:425–462, 1993.

[3] C.L.Fullmer and J.Garcia-Luna-Aceves. Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks. In *Proceedings of ACM SIGCOMM '95*, 1995.

[4] Norival R. Figueira and Joseph Pasquale. Remote-Queueing Multiple Access (RQMA): Providing Quality of Service for Wireless Communications. In *Proc. IEEE INFOCOM'98*. IEEE Computer Society, April 1998.

[5] Ching-Chih Han, Chao-Ju Hou, and Kang G. Shin. On Slot Allocation for Time-Constrained Messages in Dual-Bus Networks. *IEEE Trans. on Computers*, 46(7):756–767, July 1997.

[6] Ching-Chih Han, Kwei-Jay Lin, and Chao-Ju Hou. Distance-Constrained Scheduling and Its Applications in Real-Time Systems. *IEEE Trans. on Computers*, 45(7):814–826, July 1996.

[7] C. L. Liu and James W. Layland. Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. *J. of ACM*, 20(1):46–61, 1973.

[8] S. Nanda, D.J. Goodman, and U. Timor. Performance of PRMA: A Packet Voice Protocol for Cellular Systems. *IEEE Trans. Veh. Techn.*, 40:584–598, 1991.

[9] Abhay K. Parekh and Robert G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case. *IEEE/ACM Trans. on Networking*, 1(3):344–357, June 1993.

[10] Xiaoxin Qiu and Victor O.K. Li. Dynamic Reservation Multiple Access (DRMA): A New Multiple Access Scheme for Personal Communication System (PCS). *Wireless Networks*, 2:117–128, 1996.

[11] N.D. Wilson, R. Ganesh, K. Joseph, and D. Raychaudhuri. Packet CDMA Versus Dynamic TDMA for Multiple Access in An Integrated Voice/Data PCN. *IEEE JSAC*, 11:870–884, 1993.

[12] Yi Ye, Chao-Ju Hou, and Ching-Chih Han. QGMA: A New MAC Protocol for Supporting QoS in Wireless Local Area Networks. Technical Report, Dept. of Electrical Engineering, The Ohio State University, 1998.