Scheduling Multimedia Services in a Low-Power MAC for Wireless and Mobile ATM Networks

Jyh-Cheng Chen, *Member, IEEE,* Krishna M. Sivalingam, *Member, IEEE,*
Prathima Agrawal, *Fellow, IEEE,* and Raj Acharya

Abstract— **This paper describes the design and analysis of the scheduling algorithm for energy conserving medium access control (EC-MAC), which is a low-power medium access control (MAC) protocol for wireless and mobile ATM networks. We evaluate the scheduling algorithms that have been proposed for traditional ATM networks. Based on the structure of EC-MAC and the characteristics of wireless channel, we propose a new algorithm that can deal with the burst errors and the location-dependent errors. Most scheduling algorithms proposed for either wired or wireless networks were analyzed with homogeneous traffic or multimedia services with simplified traffic models. We analyze our scheduling algorithm with more realistic multimedia traffic models based on H.263 video traces and self-similar data traffic. One of the key goals of the scheduling algorithms is simplicity and fast implementation. Unlike the time-stamped based algorithms, our algorithm does not need to sort the virtual time, and thus, the complexity of the algorithm is reduced significantly.**

Index Terms—**Low-power operation, multiple access methods, queuing and scheduling algorithms, wireless and mobile ATM, wireless multimedia communications.**

I. Introduction

THIS paper describes the design and analysis of a scheduling algorithm for a low-power medium access control (MAC) protocol for wireless/mobile ATM networks. The design of the protocol — denoted EC-MAC (energy conserving medium access control) — is driven by two major factors. The first factor is that the access protocol should be energy-efficient since the mobiles typically have limited power capacity. The second factor is that the protocol should provide support for multiple traffic types, with appropriate quality-of-service (QoS) levels for each type. In [1] and [7], the design and analysis of EC-MAC and the comparison of energy consumption to a number of other protocols have been provided. The core of the protocol that determines the performance and guarantees the QoS is the scheduling algorithm associated with the MAC protocol. Such a scheduling algorithm is the focus of this paper. By this scheduling algorithm, we show that EC-MAC, in addition to low energy consumption, can achieve high channel utilization, low packet delay, and meet the QoS requirements for multimedia traffic.

Many queuing and scheduling algorithms have been proposed for conventional wired ATM networks. The framework is that there are queues in switches. The scheduling disciplines then schedule packets in queues accordingly. The schedule disciplines may be as simple as first-in–first-out (FIFO) or round robin or based on virtual finishing times, such as virtual clock [8], SCFQ [9], etc. Previous work for wireless ATM has reported mechanisms for providing the base station with the transmission requests [10], [11]. The scheduling algorithms, however, were not addressed in extensive detail. Recently, research on extending the scheduling algorithms proposed in wired networks to wireless domain has been reported [12], [13]. The major concern addressed here is modification of conventional wired scheduling algorithms to deal with the error-prone wireless channel. However, scheduling with low power in consideration has not been addressed so far. This paper proposes an algorithm that addresses this important issue.

Most proposed algorithms were analyzed with homogeneous traffic type or multimedia services with simplified traffic models. However, variable bit rate (VBR) traffic such as video applications based on MPEG or H-series [14], [15] coding has high variation in the number of packets generated. Those algorithms analyzed by simplified video models fail to capture the features in real applications. The proposed scheduling algorithm is studied using discrete-event simulation that uses realistic multimedia traffic models. Voice traffic is modeled by a slow speech activity detector (SAD) for talkspurts and silent gaps [16]. Data is modeled using the self-similar traffic models described in [3] and [4]. Video traffic is based on the real trace data from several H.263 [2] video sources.

Sections II and III provide overview of the EC-MAC wireless access protocol and available scheduling algorithms, respectively. Section IV discusses issues and criteria unique to scheduling in the wireless network. Section V describes the proposed algorithm. Sections VI and VII provide the performance analysis. Section VIII concludes this paper.

Manuscript received September 9, 1998; revised February 18, 1999. A preliminary version of this paper was presented at IEEE PIMRC 1998, Boston, MA. This work was supported in part by the Air Force Office of Scientific Research under Grants F-49620-97-1-0471 and F-49620-99-1-0125. The associate editor coordinating the review of this paper and approving it for publication was Dr. Hiroshi Yasuda.

J.-C. Chen is with Telcordia Technologies, Morristown, NJ 07960 USA (e-mail: jcchen@research.telcordia.com).

K. M. Sivalingam is with the School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164 USA (e-mail: krishna@eecs.wsu.edu).

P. Agrawal is with Telcordia Technologies, Morristown, NJ 07960 USA (e-mail: pagrawal@research.telcordia.com).

R. Acharya is with the Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260 USA (e-mail: acharya@cse.buffalo.edu).

Publisher Item Identifier S 1520-9210(99)04211-X.

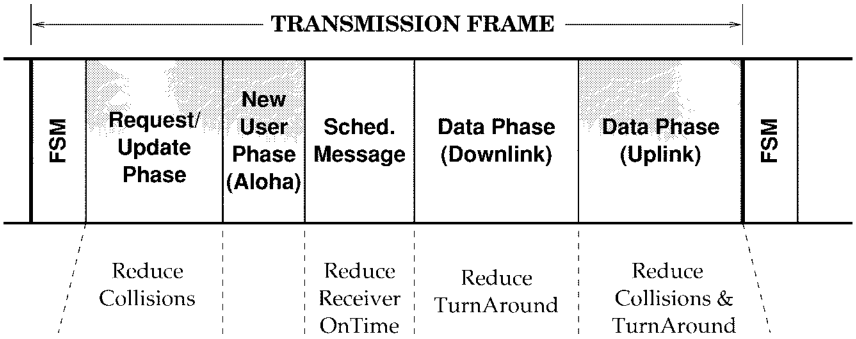


Fig. 1. Definition of the different phases in EC-MAC protocol.

II. Overview of EC-MAC

The network architecture of EC-MAC [1] is mainly derived from the SWAN network built at Bell Labs [17] — one of the first wireless ATM network testbeds. The access protocol is defined for an infrastructure network with a single base station (BS) serving mobiles in its coverage area. The goals of low energy consumption and QoS provision lead us to a protocol which is based on reservation and scheduling strategies.

Transmission in EC-MAC is organized by the BS into frames. Each frame is composed of a fixed number of slots, where each slot equals the basic unit of wireless data transmission. The frame is divided into multiple phases as shown in Fig. 1.

Frame Synchronization: At the start of each frame, the BS transmits the frame synchronization message (FSM) on the downlink. This message contains framing and synchronization information, the uplink transmission order for reservations, and the number of slots in the new user phase. From energy conservation perspective, it is desirable that the request/update phase (following this phase) should not operate in a contention mode. The request/update phase can be made collision-less by letting the BS broadcast a list containing the set of the mobile ID's. The transmission order of the ID's implicitly defines the order in which mobiles transmit their request/update information. Each mobile is allocated one slot during the request/update phase.

Request/Update: The request/update phase is composed of uplink request transmissions from the mobiles. During the uplink phase, each registered mobile transmits new connection requests and queue status of established queues according to the transmission order received in FSM.

New-User Phase: This phase allows new mobiles that have entered the cell coverage area to register with the BS. This phase is operated in a contention mode, using slotted aloha. The length of this phase is variable. The BS's FSM broadcasts the available number of slots for user registration during this phase. The BS initially starts with a small number of slots, and dynamically adjusts the number of slots based on monitoring the number of collisions. A maximum number of new-user slots is specified. The BS transmits all the acknowledgments and registration information for each mobile in a subsequent downlink message.

Schedule Message: The BS broadcasts a schedule message (SM) that contains the slot permissions for the subsequent data phase. Each permission identifies the mobile/VC combination that should transmit/receive in a given data slot. The data phase includes downlink transmissions from the BS and uplink transmissions from the mobiles.

Data Phase: Downlink transmission from the BS to the mobiles is scheduled considering the QoS requirements of the individual VC's. Likewise, the uplink slots are allocated using scheduling algorithm described below.

The above description provided the frame structure of the EC-MAC protocol. The focus of this paper is on the development and analysis of the scheduling algorithm used for allocating uplink and downlink slots. The next few sections provide more details on this.

III. Overview of Scheduling Algorithms

This section provides a brief overview of the scheduling algorithms available in the literature for both wired and wireless networks.

A. Scheduling Algorithms for Wired and ATM Networks

In order to guarantee the QoS in packet or ATM networks, proper scheduling algorithms in the routers or switches are necessary. A scheduler may be regarded as a queuing system consisting of a server providing service to a set of customers. Many service disciplines have been proposed in the literature to achieve 1) fairness in the allocation of bandwidth and 2) the bounded end-to-end delay. This section overviews algorithms provided for conventional wired networks.

Extensive research has been done for fair packet queuing and scheduling since the *weighted fair queuing* (WFQ) [18] was proposed. In [18], an idealized fluid-flow model and the packet-by-packet version (WFQ) were proposed. In WFQ, there is a virtual time as that in *virtual clock* [8] associated with each packet. Packets are sorted based on their virtual times and transmitted in that order. In [19], the idealized fluid-flow model and the packet-by-packet version are referred as *generalized processor sharing* (GPS) and *packet-by-packet generalized processor sharing* (PGPS), respectively. Several algorithms have been proposed to address its computational complexity or improve the performance of WFQ or PGPS. Algorithms such as *self-clocked fair queuing* (SCFQ) [9], *start-time fair queuing* (SFQ) [20], *worst case fair queuing* (WF^{2}Q) [21], *deficit round robin* [22], and *frame based FQ* [23] have been proposed in the literature. Recently, an algorithm called *leap forward virtual clock* (LFVC) can be implemented by "approximate sorting" and a finite-universe priority queue to achieve $O(\log\log(N))$ processing time per packet, where N is the number of concurrent flows at a router [24]. The classical virtual clock and SCFQ can be thought of as special cases of LFVC by setting the leap forward parameter appropriately.

Most of the previous research on integrated services networks has focused on guaranteeing QoS for each *individual session*, and recent work has argued that it is also important to support a hierarchical link-sharing service, for example, *class-based queuing* (CBQ) [25], *hierarchical packet fair queuing* (H-PFQ) [26], and *hierarchical fair service curve* (H-FSC) [27]. With hierarchical link-sharing, there is a class hierarchy associated with each link that specified the resource allocation policy for the link. A class represents some aggregate of traffic streams that are grouped according to protocol, traffic type, or other criteria. One of the important goals of hierarchical link-sharing service is to guarantee certain minimum bandwidth for each class. The other goal is to have a proper scheme to distribute fairly the excess bandwidth unused by a class to its sibling classes. It also need to support real-time and priority services.

An overview and comparison of some of the proposed algorithms can be found in [5]. It discusses the issues and tradeoffs in designing service disciplines to provide guaranteed performance service in packet-switching networks. In this paper, a service discipline can be classified as either *work-conserving* or *nonwork-conserving*. In a work-conserving discipline, a server is never idle when there is a packet to transmit. Nonwork-conserving disciplines, however, allow the server to be idle if no packet is eligible to be transmitted. Delay earliest-due-date (Delay-EDD) [28], head-of-the-line earliest-due-date (HOL-EDD) [29], virtual clock [8], WFQ [18], PGPS [19], SCFQ [9], and WF^{2}Q [21] are classified as work-conserving disciplines. Examples of nonwork-conserving disciplines are jitter earliest-due-date (Jitter-EDD) [30], stop-and-go [31], hierarchical round robin [32], and rate-controlled static priority (RCSP) [33].

Based on the internal architecture, the service disciplines can be also classified as *frame-based* or *sorted priority* [6]. In a frame-based algorithm, there are frames with either fixed or variable length in time domain. Reservations for each session are made to determine the maximum amount of traffic the session can transmit in each frame. In a sorted-priority scheme, there is a *virtual time* — a global variable — associated with the outgoing link being scheduled. When a packet arrives, the virtual time associated with this packet is computed. Packets in the queue are then sorted in the increasing order of the virtual time and transmitted in this order. Based on the implementation of the virtual time function, there are various different algorithms such as WFQ, SCFQ, SFQ, WF^{2}Q, etc. listed above. Fair packet queuing algorithms differ primely in two aspects: the computation of the system virtual time function and the packet selection policy. The hardware implementation issues of fair queuing algorithms in ATM networks are addressed in [6].

B. Scheduling Algorithms for Wireless Networks

While *fair queuing and scheduling* has been extensively researched for conventional wired packet- or cell-based networks, the research for wireless fair queuing and scheduling has not been addressed too much in the literature. Most of the work in wireless networks has been performed from the perspective of wireless channel access control, where the emphasis has been on the mechanism of MAC once the scheduling algorithm has been worked out [10], [11].

Intuitively, all of the fair packet queuing/scheduling and link-sharing algorithms will work well in wireless channel. However, there are two key characteristics of wireless channel that make the fair packet queuing model inapplicable: 1) bursty channel errors and 2) location-dependent channel capacity and errors [12]. In wireless networks, the errors are often bursty. The continuous scheduling and transmission for mobiles in a burst error state will cause poor channel utilization and high energy consumption. The errors are also location-dependent which means only part of the mobiles can transmit. Therefore, only subset of queues should be scheduled.

Channel state dependent packet (CSDP) scheduling was proposed to address bursty channel errors in wireless LAN's [34]. However, it does not address the issues of fairness, throughput, and delay guarantees. In [12], authors proposed a fair scheduling algorithm for wireless packet networks. It gives credits to mobiles in bursty errors rather than schedule bandwidth for them. BS then compensates all the credits later to guarantee the fairness. Because allowing an unbounded amount of credits can result in denial of service to other sessions, only bounded amount of credits is allowed. However, this credit adjustment scheme may not be useful for real-time applications. For example, the compensation of late packets for a video conference might be meaningless since the late packets will be dropped rather than be played back. The other algorithm that combines CBQ [25] with CSDP [34] was proposed in [35]. In this paper, authors proposed an enhanced CBQ for multiple classes of traffic with CSDP to improve channel utilization. The focus of this work is majorly on fairness and throughput. It, however, does not analyze the proposed scheme with other QoS parameters with diverse traffic types. The scheduling of VBR video with different rates at different times is not addressed. Reference [13] also defined a fair scheduling algorithm for wireless networks. A protocol called RQMA, which has virtual clock as the scheduling discipline for wireless ATM, was proposed in [36]. Similarly, [37] proposed an algorithm based on SCFQ for wireless ATM. An algorithm based on GPS model for hybrid CDMA/TDMA was also proposed in [38]. However, none of them concerns about the energy consumption. In addition, realistic multimedia traffic models were not used to investigate the performance. It is uncertain how well these algorithms will be for multimedia terminals with limited battery power.

IV. Desired Scheduling Algorithm Features

As discussed above, our goal is to design a scheduling algorithm for EC-MAC (a low-power MAC protocol) for wireless ATM networks supporting multimedia traffic and diverse QoS guarantees. To meet the requirements addressed above, the following are preferable features in the scheduling algorithms.

1) *Coarse-Grained Scheduling:* The computation and broadcast of the schedule is done on a slot-by-slot basis in most algorithms such as [10], which is good for fine-grained scheduling. However, packet-by-packet announcement incurs too much energy consumption since the receiver has to be turned on during every slot. In a TDMA system, the scheduling should be frame-based rather than slot-based.
2) *Contiguous Bandwidth Allocation:* Scheduling algorithms should allocate contiguous slots to each mobile to conserve energy. Therefore, each mobile only needs to listen to the announcement once and turn on the transceiver for all packets. This will be particularly beneficial for mobiles with low battery power. This can reduce the number of transmitter/receiver turn-around time as well.
3) *Centralized Algorithm:* A distributed algorithm where each mobile computes the schedule independently may not be desirable because it may not receive all the reservation requests due to radio and error constraints. Besides, schedule computation consumes power and is thus better relegated to the BS. This suggests that a centralized scheduling mechanism will be more energy efficient.
4) *Dealing with Errors:* In high-speed networks based on fiber optics, errors are limited and random in nature. In wireless networks, errors are bursty and location dependent [34]. At a time, not all mobiles are ready for packet transmission/reception due to the location-dependent errors. In addition, errors in each mobile are bursty due to fading and interference. Therefore, the algorithm should consider errors and allotted slots not being used by the mobiles.
5) *Diverse Traffic Types:* Video applications generate variable amount of traffic at different times. The simplified traffic models fail to capture the features in real applications. In order to get maximum multiplexing gain while also maintain the QoS for each session, scheduling algorithms should dynamically adjust the bandwidth allocated to VBR services.
6) *Multiplexing Gain:* One of the desired features of ATM networks is the multiplexing gain. By multiplexing VC's, we can get better channel utilization. This is more important in wireless networks because of the limited bandwidth. To get more multiplexing gain, boundaries between different traffic may not be the choice. Boundaries between traffic types may result in noncontiguous bandwidth allocation as well.
7) *Fairness in Scheduling:* In order to ensure fairness, scheduling algorithms may need to reimburse bandwidth for available bit rate (ABR) and (UBR) unspecified bit rate traffic for those mobiles in error state to guarantee long-term fairness. Since there are deadlines for each packet/cell in real-time applications, this reimbursement may not be necessary.
8) *Connection Admission Control (CAC):* In [5], the author pointed out that different service disciplines need different admission control algorithms. A complete solution needs to specify both the service discipline and the associated connection admission control conditions. CAC might want to admit more sessions to get maximum multiplexing gain. However, admitting too many sessions will affect the QoS of the existent sessions. Therefore, a good CAC algorithm should accompany with the scheduling algorithm.
9) *Simplicity and Low Complexity:* For easy implementation and good performance, a simple but efficient algorithm is necessary. Number of sorting and searching in the algorithm should be limited.
10) *Hybrid TDMA/CDMA:* Most scheduling algorithms proposed so far are based on TDMA systems. Although an algorithm based on GPS model for hybrid CDMA/TDMA was proposed in [38], energy consumption base on battery power is not a major concern here. For a hybrid TDMA/CDMA system, power control algorithms should be included in the scheduling algorithms to further reduce the energy consumption for low-power users [39].

V. Proposed Algorithm

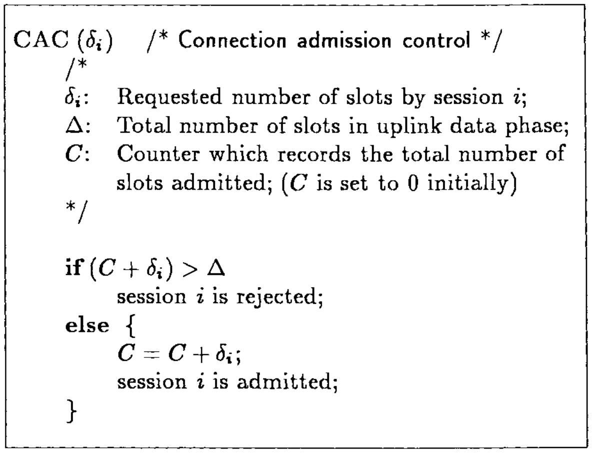
We proposed a *priority round robin with dynamic reservation update and error compensation* scheduling which deals with the QoS guarantees for multimedia services. Low-power operation in this algorithm is done by contiguous bandwidth allocation and by cooperating with EC-MAC.

A. Connection Admission Control (CAC)

After a mobile is admitted to this cell — either locally generated or handed off, it may request bandwidth for several VC's as they are created. The CAC's goal is to maintain QoS for all existent VC's while admitting new VC's.

In [40], a measurement-based admission control for VBR video was proposed, where a new session is admitted based on measured utilization. In wireless environments, the measured utilization may not be the actual load of the system because some mobiles might be in error states. The measured utilization may fail to take those mobiles into account. Hence, in a wireless system, this algorithm may admit more traffic than the system can afford.

We use a simple algorithm in which each VC sends the minimum guaranteed number of slots it needs as part of session set up. A counter is used to record the total number of slots which have been admitted. If the counter exceeds the number of slots in uplink data phase after adding this VC's request, the VC is rejected. Otherwise, it is admitted. The counter is incremented by the number of this VC's request slot(s). By this admission control algorithm, the total rate of all admitted VC's is always less than or equal to the maximum capacity in data phase. Therefore, the QoS of existing VC's will not be affected by the newly admitted VC. The algorithm is shown in Fig. 2.



```
CAC ($\delta_i$)   /* Connection admission control */
    /*
    $\delta_i$:  Requested number of slots by session $i$;
    $\Delta$:  Total number of slots in uplink data phase;
    $C$:  Counter which records the total number of
          slots admitted; ($C$ is set to 0 initially)
    */

    if $(C + \delta_i) > \Delta$
        session $i$ is rejected;
    else {
        $C = C + \delta_i$;
        session $i$ is admitted;
    }
```

Fig. 2. Connection admission control algorithm.

B. Scheduling

The proposed algorithm performs *coarse-grained scheduling* based on the frame structure of EC-MAC. Although many algorithms have been proposed for conventional wired ATM networks, most of them are based on packet-by-packet scheduling which are good for *fine-grained scheduling* only. For example, algorithms based on time-stamp such as virtual clock [8] and SCFQ [9] are not applicable to EC-MAC because they need to know the arrival time of each packet. Other type of algorithms such as HOL-EDD [29] might be modified for frame-based scheduling. Since it does not allow a session to be served at different rates at different times, a VBR video session cannot improve the delay performance without requesting the peak bandwidth. A *multirate* service algorithm was proposed to address the scheduling of VBR video [40]. However, this algorithm is fine-grained based on time-stamped priority. In addition, it was proposed for high-speed networks where errors are negligible. In wireless domain, algorithms such as that proposed in [12] do not consider the energy consumption factor. In addition, realistic multimedia traffic models are not used to investigate the performance.

The proposed algorithm is a *priority round robin with dynamic reservation update and error compensation* scheduling. The scheduler is currently defined to handle constant bit rate (CBR), e.g., voice, variable bit rate (VBR) e.g., video, and unspecified bit rate (UBR) e.g., data, traffic. The scheduler gives higher priority to CBR and VBR traffic. These traffic sources can make requests for slot reservations that will be satisfied by the scheduler. UBR traffic, on the other hand, is treated with low priority and without reservation. Within the same traffic type, the different connections are treated using round robin mechanism.

The BS maintains two tables: *request table* and *allocation table*. The request table maintains the queue size of the VC of each mobile, the error state of the mobile, the number of requested reservations for CBR and VBR traffic, and the number of credits for UBR traffic. The purpose of the allocation table is to maintain the number of slots scheduled for each VC and each mobile. This table is essentially broadcast as the schedule to the mobiles. Based on this table, the BS allocates contiguous slots within a frame for each mobile, which will be discussed later.

The BS first allocates slots to CBR VC's which have been currently admitted. Because of the connection admission control (CAC) described above, CBR VC's that belong to mobiles in nonerror (good) states are satisfied with their required rates (please refer to Section VI for formal proof). The CBR VC's are allocated X slots every Y frames, based on the traffic requirements. For instance, with a 12-ms TDMA frame, a 32-Kbps voice source is allocated one 48-byte slot per frame.

For sources with VBR traffic, the BS maintains the number of slots allocated in the previous frame. Let the current request of source i be C_i slots and the allocation in the previous frame be P_i slots. If $C_i < P_i$, C_i slots are allocated, and the remaining $P_i - C_i$ are released. If $C_i > P_i$, P_i slots are allocated in the first round. In the second round, extra slots available are evenly distributed among the VBR sources whose requests have not been fully satisfied in the first round.

Since there is correlation in a VBR video source, the reservation in current frame period represents the prediction for next frame. By the adjustment, the bandwidth allocation in each frame is different depending on the current traffic load and the number of packets generated by VBR sources. The reservation, hence, is updated dynamically in each frame for VBR traffic.

The BS then schedules UBR traffic after the scheduling of CBR and VBR. If the mobile is in error state, the BS adds credit(s) in the corresponding entry in request table. Otherwise, the BS either schedules slot(s) to this VC or schedules the aggregate credits this VC has until there is no more slot available. The reason for this credit is to ensure long-term fairness. This credit adjustment scheme is not applied to voice and video traffic since late packets will be dropped rather than be played back in such applications.

C. Contiguous Bandwidth Allocation

The scheduling algorithm described above does not really announce the slot allocation for each VC. Instead, it only updates the corresponding entry in allocation table. The allocation table can be implemented as a two-dimensional (2-D) array with one dimension for each mobile and the other dimension for each VC in this mobile. For example, each mobile has one entry which has three fields for three VC's in allocation table. The BS broadcasts the slot id and the number of slots for each VC by looking at the entry of each mobile. The pseudo code is listed in Fig. 3. By this allocation table, each mobile listens to all schedule beacons destined to it contiguously. It also gets slot allocation in data phase contiguously for all different traffic types although the scheduling is done on a traffic type basis. Therefore, mobiles only need to turn on transmitter/receiver once in schedule phase and data phase. Please note the total number of allocated slots in allocation table is less or equal to the total slots in data phase. This has been checked in the scheduling algorithm described above. As stated earlier, we want to have a fixed-size schedule beacon so mobiles know when to turn on receivers in

```
BROADCAST_SCHED ()   /* Broadcast schedule message based on allocation table */
    /*
    $N_m$:                  Total number of mobiles in the system;
    $N_{vc}$:               Maximum number of VCs in each mobile;
    slod_id:                Beginning slot in data phase for each VC;
    Sched [ ]:              One-dimension array for each schedule beacon;
    Allocate [ ] [ ]:       Two-dimension array of the allocation table ;
    */

    $index = slot\_id = 0$;
    for ($i = 0$; $i < N_m$; $i++$)      /* for each mobile in the array */
        for ($j = 0$; $j < N_{vc}$; $j++$)      /* for each VC in the mobile */
            if (Allocate [i] [j] != 0) {      /* if the entry is not zero */
                /* announce the slot allocation in schedule beacon */
                Sched [index] .macid.id = i;
                Sched [index] .macid.vc = j;
                Sched [index] .slot_id = slot_id;      /* beginning slot in data phase */
                Sched [index] .slot_num = Allocate [i] [j];      /* number of allocated slot(s) */

                slot_id += Allocate [i] [j];      /* increment the slot id */
                index++;      /* next schedule beacon */
            }
```

Fig. 3. Contiguous bandwidth allocation.

schedule phase. In current implementation, we have a small-size schedule beacon for each VC. Mobiles may need to listen to several schedule beacons for all different VC's, in which all of them have the same mobile id. We can have a bigger schedule beacon which has more fields containing information of all VC's for each mobile. Hence, each mobile only needs to listen to one beacon. However, not all VC's will request resources if the traffic load is light. This bigger beacon may cause waste due to some empty fields. The tradeoff in the small-size and big-size schedule beacon is that bandwidth may be wasted in the small-size beacon when the traffic load is heavy because of the replication of mobile id in each beacon, whereas the big-size beacon will cause waste when the traffic load is light.

By the algorithm in Fig. 3 and the allocation table, the slot allocation is announced on frame basis rather than on slot basis. Mobiles also only need to turn on the transceiver once for all different types of packets.

D. Dealing with Errors

This section describes how the scheduling algorithm deals with bursty and location-dependent errors. At a time, only some of the mobiles may be capable to communicate with the BS—the others might be in error state. Since a mobile may encounter errors during any phase of the time frame, we discuss them individually as follows.

1) If a mobile is in error state during BS frame synchronization message (FSM) reception, it will not receive its transmission order. Thus, it will not send the request in the uplink of request/update phase, and the BS will mark the mobile as in error state. The scheduling algorithm might assign credits to the mobile depending on the traffic type. In case the mobile changes to good state any time after this phase, the mobile will not be able to transmit in the subsequent data phase. It could decide to receive broadcast packets.
2) If errors happen during the uplink of request/update phase, the BS will mark the mobile as in error state because it does not receive the transmission request. When the mobile sends request in the subsequent request/update phase, BS will mark the mobile as in good state. The situation is similar to the one above.
3) If errors happen while a mobile is receiving the schedule message, bandwidth that has been scheduled to this mobile will not be utilized. This loss is limited to only one data phase which is typically smaller than the average burst error length of 100 ms [34]. The BS will mark the mobile as in error state when it does not receive this mobile's data during the scheduled uplink slots. The BS will mark the mobile as in good state when it receives the requests from mobiles in request/update phase again.
4) If errors happen during the downlink data phase or the mobile does not turn on receiver because of missed schedule message, the BS will hold packets until the corresponding mobile returns back to good state. Mobiles can acknowledge the packets they receive when they send requests in next request/update phase. Thus, the BS can know whether mobiles have received the downlink packets or not. The BS deletes packets from queues only after it receives acknowledgments.
5) If errors happen during the uplink of the data phase, the BS will not receive the packets sent from the mobiles in error state. The BS acknowledges the packets it received in the next FSM. Mobiles delete packets from queues only after receiving acknowledgments or the deadline of real-time packets has expired. The BS will know the

actual queue size of each VC and reschedule the packets when it receives the requests from mobiles in uplink request/update phase again.

This section described the mechanisms defined in EC-MAC to handle bursty and location-dependent errors during the various phases. The following sections provide the properties of the new approach and a simulation based performance analysis.

VI. Properties of the New Approach

The analysis of most proposed algorithms mainly focuses on the throughput fairness and the end-to-end delay bound. The fairness is to ensure that each session can get the guaranteed rate it needs, and the unused bandwidth is distributed evenly to all sessions. Service fairness index (SFI) [9] and worse-case fairness index (WFI) [21] are two commonly used fairness measures.

For wired networks, the scheduling algorithm is implemented in each switch, and the algorithm can work independently. In wireless networks, many of them are implemented in the last hop only [17], and the scheduling algorithms must cooperate with the MAC protocols. Our scheduling algorithm is designed for EC-MAC, which is used for wireless infrastructure networks. Therefore, our analysis is based on the last hop only, i.e., the wireless links, and focuses on the uplink transmission, i.e., mobiles to BS.

In analysis, we focus on the rate guarantee, delay bound, jitter bound, and loss bound. As argued in above sections, the fairness might not be very important for multimedia traffic. Once a session can get the guaranteed rate and fulfill the QoS such as delay and loss bounds it needs, how the bandwidth is distributed might not be too important. In our algorithm, each session can get its required bandwidth, and the unused bandwidth is used mainly to maintain the different QoS requirements for VBR VC's because they may generate much more packets than they set in CAC. We also discuss the time complexity of the algorithm. Since our algorithm does not need to time stamp each packet then sort all of them, the scheduling decision can be done in $O(1)$. The contiguous bandwidth allocation algorithm can be done with $O(V)$, where V is the total number of VC's in all mobiles.

Lemma 1: Let N be the number of admitted VC sessions. Let Δ be the total number of slots in the uplink data phase in a time frame, and δ_i be the number of slots session i requests during CAC. With the CAC algorithm described in Fig. 2, we have $\sum_{i=1}^{N} \delta_i \leq \Delta$.

Proof: In the CAC algorithm, there is a counter that records the total number of slots used for admitted sessions. A new session is admitted only when the available slots are larger than the request. From the pseudo code listed in Fig. 2, we can see that $\sum_{i=1}^{N} \delta_i \leq \Delta$ after admitting N sessions.

Theorem 1—CBR Rate Guarantee: By the scheduling algorithm, all CBR VC's that belong to mobiles in nonerror states and conform to the rate set in CAC are satisfied with their requested rates.

Proof: Let ζ be the set of CBR VC's that belong to mobiles in nonerror states and conform to the rate set in CAC. If some of them cannot be satisfied with the rates set in CAC, then

$$\sum_{i\in\zeta} \delta_i > \Delta. \tag{1}$$

Let N_{CBR} be the number of CBR sessions in ζ, i.e., $N_{\text{CBR}} = |\zeta|$, in which $N_{\text{CBR}} \leq N$. Because CBR sessions are scheduled with highest priority among other traffic types, by Lemma 1, we get

$$\sum_{i\in\zeta} \delta_i \leq \sum_{i=1}^{N} \delta_i \leq \Delta. \tag{2}$$

Equation (1) conflicts with (2). Therefore, all VC's in ζ can be satisfied with the rates set in CAC.

Definition 1—Delay: The packet delay is defined as the time elapsing between packet arrival and departure in a mobile.

Theorem 2a—CBR Delay Bound: The maximum packet delay of CBR sessions which belong to mobiles in nonerror states and conform to the rates set in CAC is $\tau + (T_2 - t_1)$, where τ is the duration of a time frame, and T_2 and t_1 are time stamps defined in Fig. 4.

Proof: Let a and d be the arrival time and departure time of a packet in CBR sessions. By Theorem 1, all CBR VC's that belong to mobiles in nonerror states and conform to the rates set in CAC can be satisfied with their requested rates. Therefore, a packet which arrives before the request/update phase will be scheduled to be transmitted in the following uplink data phase. That is, if $a \in [t_1, t_3)$, $d \in [t_4, T_5)$. Therefore, the maximum delay equals $(T_5 - t_1)$. From Fig. 4, $(T_5 - t_1)$ is equal to $\tau + (T_2 - t_1)$.

Theorem 2b—CBR Delay Bound: The maximum packet delay of CBR sessions which belong to mobiles in error states is $\min\{n\tau + (T_2 - t_1), D_c\}$, where $n \in \{1, 2, 3, \cdots\}$. τ is the duration of a time frame. D_c is the maximum packet delay for a real-time CBR session. n depends on the duration of errors.

Proof: As shown in Theorem 2a, if $a \in [t_1, t_3)$, the departure time d can be in the following uplink data phase. If errors occur during request/update phase, schedule message, or uplink data phase, this packet will not be transmitted until next frame that has no errors in such phases. If errors occur in new user phase or downlink data phase, the packet still gets chance to be transmitted in the following uplink data phase. Therefore, between $(T_2 - t_1)$ and the frame in which the packet is transmitted successfully, there might be several frames in which this packet cannot be transmitted. Thus, the maximum delay equals $(T_2 - t_1) + n\tau$, where $n \in \{1, 2, 3, \cdots\}$. The value of n depends on the interval of errors.

For some real-time CBR services, there might be a maximum tolerable delay for each packet. If the packet is not transmitted before its deadline, it is dropped. Let D_c be the maximum packet delay for a real-time CBR session. All packets have maximum delays less than D_c. The delay bound, therefore, is $\min\{n\tau + (T_2 - t_1), D_c\}$.

Theorem 2—CBR Delay Bound: The maximum packet delay of CBR sessions is

$$\min\{n\tau + (T_2 - t_1), D_c\} \tag{3}$$

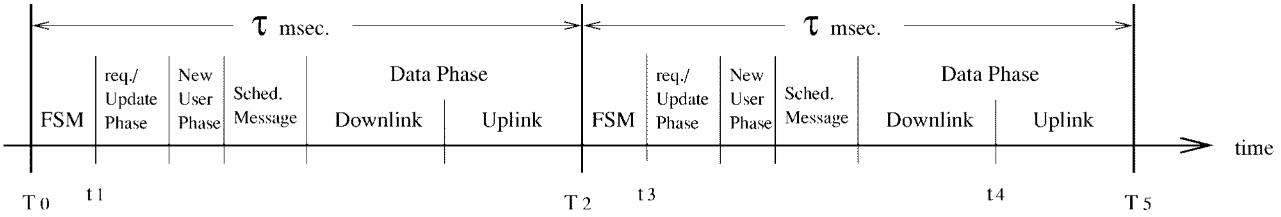


Fig. 4. Timing in the frames.

where $n \in \{1, 2, 3, \cdots\}$. n depends on the duration of errors. All other terms are same as those defined before.

Proof: The proof of this theorem can be done by combining Theorems 2a and 2b above.

Definition 2—Jitter: Jitter is defined as the variation in packet delays.

Theorem 3—CBR Jitter Bound: The jitter bound (maximum variation in packet delays) of CBR sessions is

$$\min\{n\tau + (T_2 - t_1),\, D_c\} - (t_4 - t_3) \tag{4}$$

where $n \in \{1, 2, 3, \cdots\}$. D_c, t_1, T_2, t_3, and t_4 are defined as those in Theorems 2a and 2b.

Proof: From Theorems 2a and 2b, we know that the minimum delay for any CBR sessions either in error or nonerror mobiles is $(t_4 - t_3)$. The maximum possible delay for any CBR sessions is $\min\{n\tau + (T_2 - t_1),\, D_c\}$, $n \in \{1, 2, 3, \cdots\}$ by Theorem 2. Thus, the jitter bound is the difference between the maximum and minimum delays.

Lemma 2—CBR Packet Loss Bound: In the proposed scheduling algorithm, there is no packet loss caused by the scheduling algorithm for CBR sessions conforming to the rates set in CAC. All packet losses are caused by the channel errors, and the packet lost rate is close to the channel error rate.

Proof: Suppose that some packets are lost due to the scheduling algorithm in some CBR sessions that conform to the rates set in CAC. That is, some packets in those CBR sessions in nonerror mobiles cannot be transmitted before their deadlines. However, Theorem 2a shows that the maximum packet delay is $\tau+(T_2-t_1)$ for those CBR sessions in nonerror mobiles that are generally less than the maximum packet delay D_c. Theorem 2b shows that the maximum delay may exceed D_c only when there are errors, but these packets will be dropped. Therefore, no packet is lost due to the scheduling algorithm in CBR sessions that conform to the rates set in CAC. All packet losses are caused by the channel errors.

Because the packet loss is caused by the channel errors, the packet loss rate should be equal to the error rate. However, there are some different cases when errors occur in different phases in a frame as discussed in Section V-D. There may be a case that CBR packets are generated and transmitted just when there is no channel error, although errors do occur in other time phases. Thus, the packet loss rate is less than the channel error rate. On the other hand, it is also possible that errors happen in request/update phase or schedule message only, but many packets will not be transmitted in the following data phase. Hence, the packet loss rate may be larger than the channel error rate. As discussed in Section V-D, these are extreme cases. The packet loss rate, therefore, is generally close to the channel error rate.

Lemma 3—VBR Rate Guarantee: Let Δ be the total number of slots in the uplink data phase in a time frame. Δ_{VBR} is the total number of slots available for VBR sessions in the uplink data phase in a frame. Let ϑ be the set of VBR VC's that belong to mobiles in nonerror states, and σ_i be the number of slots that VBR session i requests in CAC. After some VBR VC's request bandwidth in CAC successfully, we have $\Delta \ge \Delta_{\mathrm{VBR}} > 0$ and $\Delta_{\mathrm{VBR}} \ge \sum_{i\in\vartheta} \sigma_i$.

Proof: After some VBR VC's request bandwidth in CAC successfully, there must be some bandwidth available for VBR sessions according to the scheduling algorithm. Therefore, $\Delta_{\mathrm{VBR}} > 0$. In the uplink data phase in a time frame, the number of slots available for VBR sessions will not exceed the total number of slots. Thus, $\Delta \ge \Delta_{\mathrm{VBR}}$.

When there is no error, the slots available for VBR sessions equal the slots reserved for VBR sessions in CAC. When some mobiles are in error states, some CBR and VBR VC's may not be able to utilize their reserved bandwidth. According to the scheduling algorithm, those unused bandwidth may be allocated to VBR VC's dynamically. Therefore, the slots available for VBR VC's may be greater than the total number of slots set in CAC. Hence, $\Delta_{\mathrm{VBR}} \ge \sum_{i\in\vartheta} \sigma_i$.

Theorem 4a—VBR Delay Bound: The maximum packet delay of VBR sessions which belong to mobiles in nonerror states is

$$\begin{cases} (T_2 - t_1) + \tau, \\ \quad \text{if } \Delta_{\mathrm{VBR}} \ge \displaystyle\sum_{i\in\vartheta} M_i \\ \min\left\{ (T_2 - t_1) + \tau \left(1 + \left\lceil \dfrac{\displaystyle\sum_{i\in\vartheta} M_i - \Delta_{\mathrm{VBR}}}{\Delta_{\mathrm{VBR}}} \right\rceil \right),\, D_v \right\} \\ \quad \text{if } \Delta_{\mathrm{VBR}} < \sum_{i\in\vartheta} M_i \end{cases} \tag{5}$$

where τ is the duration of a time frame, and T_2 and t_1 are time-stamps defined in Fig. 4. M_i is the maximum number of slots that VBR session i requires in a frame period, i.e., it generates packets in peak rate. $\lceil\ \rceil$ is the ceiling function.

Proof: In CAC, each VBR VC sets the average rate it requires. However, the number of packets generated in each VBR session is variant in time. If the available slots, Δ_{VBR}, in each frame are enough for $\sum_{i\in\vartheta} M_i$, the total number of backlogged VBR packets in nonerror mobiles will be scheduled. When each VBR session generates maximum number of packets, all VBR packets generated before request/update phase can be transmitted in the following uplink data phase.

Therefore, the maximum delay is $(T_2 - t_1) + \tau$. On the other hand, if $\Delta_{\rm VBR} < \sum_{i\in\vartheta} M_i$, a packet may need to wait for some more frames, each with τ units of time, to be transmitted. The maximum number of frames a packet may need to wait is $(\lceil(\sum_{i\in\vartheta} M_i - \Delta_{\rm VBR})/\Delta_{\rm VBR}\rceil)$. Therefore, the maximum delay is $(T_2 - t_1) + \tau(1 + \lceil(\sum_{i\in\vartheta} M_i - \Delta_{\rm VBR})/\Delta_{\rm VBR}\rceil)$. As that defined in CBR, a VBR application may have a maximum tolerable delay for each packet which is set as D_v. Packets will be dropped if they are not transmitted before their deadlines. Hence, the maximum delay is as that in (5).

Please note that $\Delta_{\rm VBR}$ may not be a constant in each frame. When some CBR VC's do not utilize their reserved slots due to errors, these slots may be used by VBR VC's. Because the errors are bursty and the length of a frame is typically much smaller than the length of a burst error, the difference for $\Delta_{\rm VBR}$ in each frame is very small. Since we also take the ceiling function, the difference due to the variance in $\Delta_{\rm VBR}$ is almost negligible. By Lemma 3, we also know that $\Delta_{\rm VBR}$, the denominator, in (5) is not 0.

Theorem 4b—VBR Delay Bound: The maximum packet delay of VBR sessions that belong to mobiles in error states is

$$\begin{cases} \min\{(T_2 - t_1) + n\tau,\, D_v\} \\ \quad \text{if } \Delta_{\rm VBR} \ge \sum_{i\in\vartheta} M_i \\ \min\left\{(T_2 - t_1) + n\tau + \tau\left(\left\lceil\dfrac{\sum\limits_{i\in\vartheta} M_i - \Delta_{\rm VBR}}{\Delta_{\rm VBR}}\right\rceil\right), D_v\right\} \\ \quad \text{if } \Delta_{\rm VBR} < \sum\limits_{i\in\vartheta} M_i \end{cases} \tag{6}$$

where $n \in \{1, 2, 3, \cdots\}$, which depends on the duration of errors. Other terms have the same definition as in the above.

Proof: When errors occur during request/update phase, schedule message, or uplink data phase, the packet in an error mobile will not be transmitted until this mobile returns to good (nonerror) state. If errors occur in the new user phase, the packet still gets chance to be transmitted in the following uplink data phase. As discussed in Theorem 2b, this packet may need to wait for some more frames, i.e., in addition to the time in (5), there may be some extra τ units of time while the mobile is in error state. Therefore, we get (6) for VBR VC's in mobiles that transit between error state to nonerror state.

Theorem 4—VBR Delay Bound: The maximum packet delay of VBR sessions is

$$\begin{cases} \min\{(T_2 - t_1) + n\tau,\, D_v\} \\ \quad \text{if } \Delta_{\rm VBR} \ge \sum\limits_{i\in\vartheta} M_i \\ \min\left\{(T_2 - t_1) + n\tau + \tau\left(\left\lceil\dfrac{\sum\limits_{i\in\vartheta} M_i - \Delta_{\rm VBR}}{\Delta_{\rm VBR}}\right\rceil\right), D_v\right\} \\ \quad \text{if } \Delta_{\rm VBR} < \sum\limits_{i\in\vartheta} M_i \end{cases} \tag{7}$$

where $n \in \{1, 2, 3, \cdots\}$, which depends on the duration of errors. Other terms have the same definition as in the above.

Proof: By Theorems 4a and 4b, we can get the maximum packet delay for VBR sessions as that in (7).

Theorem 5—VBR Jitter Bound: The jitter bound of VBR sessions is

$$\begin{cases} \min\{(T_2 - t_1) + n\tau, D_v\} - (t_4 - t_3) \\ \quad \text{if } \Delta_{\rm VBR} \ge \sum\limits_{i\in\vartheta} M_i \\ \min\left\{(T_2 - t_1) + n\tau + \tau\left(\left\lceil\dfrac{\sum\limits_{i\in\vartheta} M_i - \Delta_{\rm VBR}}{\Delta_{\rm VBR}}\right\rceil\right), D_v\right\} \\ \quad -(t_4 - t_3), \quad \text{if } \Delta_{\rm VBR} < \sum\limits_{i\in\vartheta} M_i \end{cases} \tag{8}$$

where $n \in \{1, 2, 3, \cdots\}$, which depends on the duration of errors. Other terms have the same definition in the above.

Proof: As discussed in Theorem 3, the minimum delay for any sessions either in error or nonerror mobiles is $(t_4 - t_3)$. The maximum possible delay for any VBR sessions is defined in (7). Therefore, the jitter bound is the difference between the maximum and minimum delays as shown in (8).

Proposition 1a—VBR Packet Loss Bound: By proper setting in CAC, there can be no packet loss for VBR VC's that belong to mobiles in nonerror states.

Proposition 1b—VBR Packet Loss Bound: By proper setting in CAC, the packet loss rate for VBR VC's can be close to the channel error rate.

The VBR VC's such as video applications generate different amount of packets in different times. If each VBR VC reserves the maximum number of slots (peak rate), it requires in CAC and if there is no channel error, there always are enough slots to transmit VBR packets. However, not all slots may be utilized when VBR VC's generate less packets. This causes wasted slots. On the other hand, packets may be lost if it requests too few slots in CAC. A good scheduling algorithm should let each VBR VC set a reservation as small as possible in CAC while also maintaining a very small packet loss rate. Since there are errors in wireless environment, the packet loss is inevitable. The proposed new scheduling algorithm has packet loss rate close to channel error rate for VBR VC's. This will be shown by simulation in Section VII.

Definition 3—Work Conserving: A scheduling discipline is called work conserving if the server is never idle when there is a packet to transmit. Nonwork-conserving disciplines, however, allow the server to be idle if no packet is eligible to be transmitted.

Theorem 6—Nonwork-Conserving: The proposed scheduling algorithm is a nonwork-conserving discipline. When there is no channel error, the maximum server idle time is $(t_4 - t_1)$ when there is a packet ready to be transmitted. t_1 and t_4 are time-stamps defined in Fig. 4.

Proof: In the beginning, there is no packet in queues in mobiles, and therefore, the server in BS is idle. If a packet arrives a queue in FSM phase, it will be transmitted in the uplink data phase in the same frame if there is no channel error. Therefore, the maximum server idle time is less than τ. If no packet is generated in FSM phase, all packets that arrive after the request/update phase will be scheduled to be transmitted in the uplink data phase in the next frame. That is, if packets in mobiles are ready to be transmitted in $[t_1, t_3)$, they will be scheduled to be transmitted in $[t_4, T_5)$. The server will start to serve packets in the beginning of t_4. Therefore, the maximum server idle time is $(t_4 - t_1)$ after a packet is ready to be transmitted.

Theorem 7—Time Complexity: The proposed scheduling algorithm has time complexity $O(1)$ for scheduling. The time complexity is $O(V)$ for contiguous bandwidth allocation algorithm, where V is the total number of VC's in all mobiles.

Proof: The proposed scheduling algorithm knows which VC it should choose to schedule and how many slots it should allocate by just looking at the scheduling table. The table is set in the beginning and is updated in request/update phase. The update only takes some computations which take constant time. It is independent with the number of VC's. Therefore, the scheduling decision has time complexity of $O(1)$. There is no sorting and tag computation as in most scheduling algorithms such as SCFQ, etc.

For the contiguous bandwidth allocation algorithm in Fig. 3, it looks at each mobile and then each VC inside each mobile. If there are n mobiles in the system and v VC's in each mobile, the algorithm needs to look at $(n \times v)$ VC's. That is, the algorithm needs to look at all VC's in the system. Let V be the total number of VC's in the system which equals $(n \times v)$. The time complexity of contiguous bandwidth allocation algorithm, therefore, is $O(V)$.

VII. Performance Analysis

This section describes source traffic models, performance metrics studied, and simulation results for the algorithm described above using realistic source traffic models for video, voice, and data services. The analysis in this section assumes that all mobiles in the cell coverage area have already registered with the BS. The new-user phase in Fig. 1, therefore, is not incorporated.

The performance of the algorithm associated with EC-MAC protocol has been studied through discrete-event simulation. Simulation results have been obtained using the stochastic self-driven discrete-event models, written in C with YACSIM [41]. YACSIM is a C-based library of routines that provides discrete-event and random variate facilities. Steady-state transaction times and utilization were measured.

A. Source Models

The simulation results presented here consider three types of traffic—CBR, VBR, and UBR. Voice is modeled as a two-phase process with talkspurts and silent gaps [16]. Typically, such modeling classifies voice as VBR. We consider that the voice source generates a continuous bit-stream during talkspurts and is therefore classified as a CBR source in our scheduling. Video is considered as an example of a VBR source with variable number of cells per frame. Data generated by applications such as ftp, http, and e-mail is considered as an example of UBR traffic.

In simulation, each mobile terminal is capable of generating three different types of traffic: data, voice, and video. An idle mobile generates new voice calls and video calls with rates of λ_s, and λ_v, respectively. Data traffic is modeled as self-similar traffic with Hurst parameter of 0.9 (described below). The following paragraphs present the simulation models for data, voice, video, and error, respectively. The system parameters are summarized in Table I.

TABLE I
System Parameters Used for Simulation

Items	Value
Channel rate	10176 Kbps
TDMA frame length	12 ms
Number of slots (cells) per TDMA frame	288 slots
Number of slots (cells) per Data Phase	266 slots
Voice/Data/Video slot size	53 bytes
Average length of a voice call	3 min
Percentage of talkspurts for a voice call	36%
Percentage of silence gaps for a voice call	64%
Average length of a talkspurt	1.00 sec
Average length of a silent gap	1.35 sec
Speech coding rate	32 Kbps
Average length of a video call	5 min
Maximum video packet/cell delay	144 ms
Hurst parameter (index of self-similarity)	0.9

1) Data Model: Recently, extensive studies show that data traffic is self-similar in nature, and the traditional Poisson process cannot capture this fractal-like behavior [3]. The difference between self-similar and traditional models is that the self-similar model is long-range dependent, i.e., bursty over a wide range of time scales. Self-similar model shows that the traffic has similar statistical properties at a range of time scales: milliseconds, seconds, minutes, and hours. Long-range dependent traffic (fractional Gaussian noise) can be obtained by the superposition of many ON–OFF sources in which the ON and OFF periods have a Pareto type distribution with infinite variance [4]. In simulation, we use the strictly alternating ON–OFF sources with the same α-value for the Pareto distribution. The α value equals 1.2 which corresponds to the estimated *Hurst parameter*, the index of self-similarity, of $H = 0.9$ [4].

2) Voice Model: A voice source is modeled as a two-state Markov process representing a source with a *slow speech activity detector* (SAD) [16]. The probability that a principal talkspurt with mean duration t_1 s ends in a frame of duration τ is $\gamma = 1 - \exp(-\tau/t_1)$. The probability that a silent gap with mean duration t_2 s ends in a frame of duration τ is $\sigma = 1 - \exp(-\tau/t_2)$. Here, γ is the probability that a source makes a transition from talkspurt state to silent state, and σ is the probability that the source makes a transition from silent state to talkspurt state.

Measured values for t_1 and t_2 are 1.00 and 1.35 s [16], each with exponential distribution. This results in an average of 36% talkspurts and 64% silence gaps for each voice conversation. A voice cell is dropped if not transmitted after 36 ms. When a new voice cell arrives at a full queue, the *first* cell in the voice queue will be dropped.

3) Video Model: H.263 video [2] targets the transmission of video telephony at data rates less than 64 Kbps, which makes

TABLE II
VIDEO BIT RATES (IN KB/S)

Name	Mean rate	Max rate	Name	Mean rate	Max rate
Car phone 1	50.03	464.00	Car phone 2	40.56	409.40
Claire 1	33.79	658.00	Claire 2	23.20	550.60
Foreman 1	44.11	404.40	Foreman 2	31.82	368.80
Grandma 1	11.65	397.60	Grandma 2	8.33	370.20
Mother & daughter 1	21.82	433.80	Mother & daughter 2	15.72	391.40
Miss America 1	37.19	446.00	Miss America 2	23.87	392.60
Salesman 1	17.22	506.60	Salesman 2	12.69	464.00
Suzie 1	31.07	322.80	Suzie 2	22.88	285.20
Trevor 1	30.57	406.80	Trevor 2	24.95	366.40

it suitable for wireless communications. In simulation, we used the real trace data from several H.263 video sources [42]. Each video is coded by two different schemes. The first one has I and P frames only. The second one adds some other options such as PB-frames and advanced prediction mode. Each video runs for around 30 s. The frame rate is 25 fps for all videos. Table II shows the bit rates of the video sources. The table shows that the second coding scheme has lower bandwidth requirements for all the sources.

For a TDMA frame of length 12 ms (as used in the simulation), the mean number of video packets is around 1 ATM cell per TDMA frame and the maximum is 21 ATM cells per TDMA frame. In the simulation, we assume that the length of a video session is exponentially distributed with mean time of 5 min. This is achieved by randomly selecting different videos (since each video trace only lasts above 30 s).

4) Error Model: In high-speed networks based on fiber optics, errors are rare and random in nature. In wireless networks, errors are bursty and location-dependent. Models for a Rayleigh fading channel have been studied in [43]. Here, it has been shown that a first-order Markov model is an adequate approximation for a Rayleigh fading channel. In our studies, we use the slow and fast fading models proposed in [43]. The values of normalized Doppler bandwidth for slow fading and fast fading are 0.01 and 0.64, which correspond to users with moving speed about 1.5 and 100 km/h, respectively. Two fading margins are considered: 29.9978 and 19.9782 dB, which correspond to packet error rates of 0.001 and 0.01, respectively [44].

B. Performance Metrics Studied

The focus of the study is to understand what kind of service quality is provided by the scheduling algorithm with an increase in the number of mobiles supported. The metrics and analysis for EC-MAC based on energy consumption can be found in [7]. To this end, we define the following QoS parameters:

Voice-Cell Dropped Rate: The voice-cell dropped rate is defined as the ratio of *the number of voice cells dropped* to *the total number of voice cells generated.* It has been suggested in [45] that it should not exceed 1% since distortion will be perceptible otherwise.

Voice-Call Dropped Rate: The voice-call dropped rate is defined as the ratio of *the number of voice calls dropped* to *the total number of voice calls generated.* The acceptable value depends on the requirement of each system. In [11], the authors have considered 1% voice call blocking probability.

Video-Cell Dropped Rate: The video-cell dropped rate is defined as the ratio of *the number of video cells dropped* to *the total number of video cells generated.* The desired value depends on different coding algorithms and different type of services. For nonlayered MPEG-2 coding, [46] indicates that the packet loss ratios of 10^{-3} or greater for ATM cells are generally unacceptable. However, [47] shows that the quality could be improved when the cell loss rate is greater than 10^{-3} by macroblock resynchronization technique.

Video-Call Dropped Rate: The video-call dropped rate is defined as the ratio of *the number of video calls dropped* to *the total number of video calls generated.* The acceptable value depends on the requirement of each system. In order to maintain the quality of existent sessions, the CAC can restrict the number of video sessions because they generally require much more bandwidth.

Average Data Cell Delay: The average data cell delay is defined as *the average time a data cell transmitted* minus *the average time between data cell generation and transmission.* The desired value depends on the characteristics of different type of data services, such as file transfer or e-mail, etc.

Channel Utilization: The channel utilization is defined as the ratio of *the number of slots used for transmission* to *the total number of slots available.* Since the study is focused on how well a protocol can schedule mobiles for uplink transmission, we consider the uplink channel utilization only.

C. Simulation Results

The numerical results presented here study the maximum number of mobiles that can be accommodated with the desired QoS for voice, data, and video traffic. A channel rate of 10 Mbps has been considered. Each uplink and downlink in data phase is around 4.7 Mbps. The figures are plotted with offered load of 25% (G = 0.25) and 50% (G = 0.50) per mobile. Data traffic is modeled as self-similar traffic with Hurst parameter of 0.9 [4]. When load is 50%, the inter-arrival times of voice calls ($1/\lambda_s$) and video calls ($1/\lambda_v$) are 180 and 300 s, respectively. The average length of a voice call is 3 min; therefore, the voice traffic load is 180/(180 + 180) = 50%. The average length of a video call is 5 min, so the video traffic load is 300/(300 + 300) = 50% as well. The *packet*

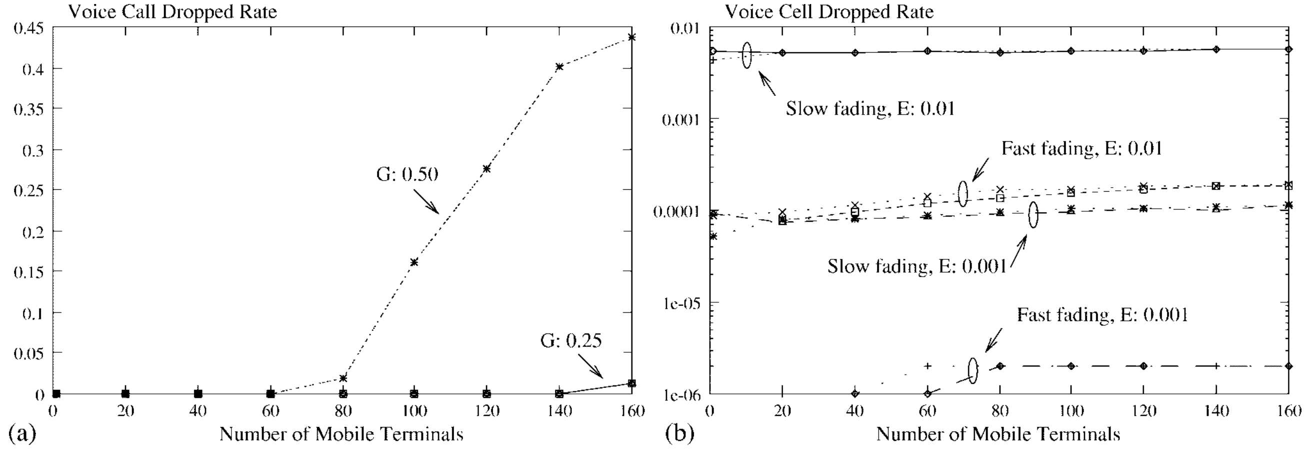


Fig. 5. Voice (a) call dropped rate and (b) cell dropped rate, where E is the packet error rate, and G is the traffic load in each mobile (channel rate: 10 Mbps).

error rates are 10^{-2} and 10^{-3} with fast fading and slow fading.

Voice-call dropped rate is considered first in Fig. 5(a), where same traffic load with different error rates and fading models leads to same results. Therefore, only two lines are visible in Fig. 5(a). As expected, less voice calls are dropped if less mobiles contend in the system. With the same number of mobiles, traffic load of 50% leads to higher dropped rate than traffic load of 25%. Fig. 5(a) also shows that the major factor for call dropped rate is traffic load rather than the fading or error rate because the call dropped rate depends mainly on CAC. When the load is 50% and the number of mobiles is greater than 80, the voice call dropped rate increases rapidly. For the load of 25%, voice call dropped rate is acceptable even when the number of mobiles is 160 regardless of the error rate.

Once a voice call is admitted, Fig. 5(b) indicates that the voice-cell dropped rate is almost independent of traffic load. In Fig. 5(b), fading and error rate are simulated for two traffic loads: 0.25 and 0.5. The cell dropped rate is less than the error rate for slow fading regardless of the number of mobiles and the load offered by each mobile. The cell dropped rate is much less than the error rate for fast fading. The difference in fast fading and slow fading is that the slow fading is more bursty than fast fading [44]. This leads to a shorter error period when error happens in fast fading. Each voice cell can tolerate 36 ms delay. Longer error period may cause more expired voice cells. Hence, slow fading has higher dropped rate. Fig. 5(b) shows that the cell dropped rate increases slightly for fast fading when the number of mobiles increases.

The CAC algorithm restricts the number of connections to maintain the QoS of admitted connections. The CAC and scheduler cooperate with each other like this: CAC deals with traffic load and scheduler deals with QoS requirements. Although cells are still dropped, that is the *de facto* nature of wireless channel due to errors. Figs. 5 and 6 show that the call dropped rate that depends on traffic load is determined by CAC, whereas cell dropped rate, that depends on fading and error rate, is determined mainly by scheduler.

Fig. 6 examines the video-call and video-cell drop rates. As discussed above, call dropped rate is determined mainly by the offered traffic load. With CAC, a video call sends the minimum guaranteed rate it needs. Based on this information, CAC decides to admit or reject this call. If a video call requests the maximum rate it needs in CAC, there will be no dropped cell ideally. However, it is wasteful to decide on admission control based on maximum bandwidth requirement. If it requests a mean rate in CAC, many other sessions can be admitted but the cell dropped rate may be unacceptable. For a H.263 video with mean rate of 1 ATM cells per TDMA frame and peak rate of 21 cells per TDMA frame, Fig. 6(a) and (b) show the results when each video sets two cells as the minimum guaranteed rate. Although the request rate set in CAC is still much less than the peak video rate, Fig. 6(b) indicates that the video-cell dropped rate is very small even with 0.01 error rate for slow fading. All others almost have 0 video-cell dropped rate in Fig. 6(b). This indicates that our *dynamic reservation update* scheme can get a good multiplexing gain. Fig. 6(a) also shows our algorithm can support 80 video sessions when load equals 0.5. More than 160 video sessions can be accepted when load equals 0.25 if the required call dropped rate is set to 1%.

Fig. 7(a) compares the data cell delay. Data traffic is transmitted when there are no other voice or video traffic pending. Although data is with lower priority and without any reservation, it still gets a chance to transmit when some voice or video sessions are in error state or when VBR video sessions generate less traffic. As expected, the data cell delay increases when the number of mobiles increases. Fig. 7(a) shows that the higher load generally has higher data delay. Channel fading and error rate, however, will not affect data delay too much. This is because the scheduling algorithm credits the error mobiles after they change back to good state.

Fig. 7(b) examines the uplink channel utilization. When the traffic load is higher, overall utilization is higher. The error rate has a little performance difference for utilization since the error rates are small compared to the total bandwidth. The channel utilization increases as the number of mobiles increases.

Although the simulation results of the delay and jitter for voice and video sources are not showed here, they do comply

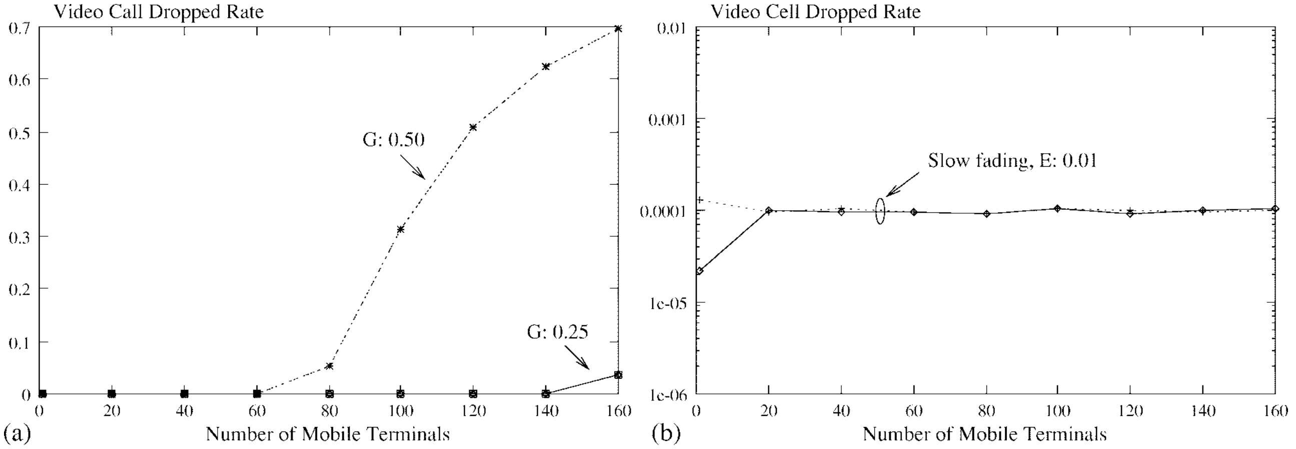


Fig. 6. Video (a) call dropped rate and (b) cell dropped rate, where E is the packet error rate, and G is the traffic load in each mobile (channel rate: 10 Mbps).

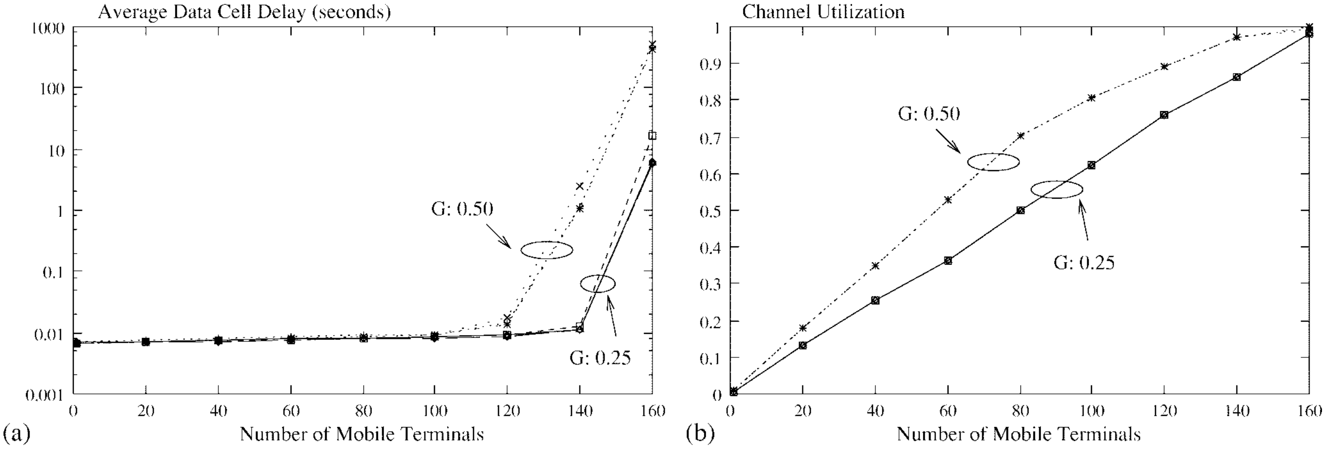


Fig. 7. (a) Data cell delay and (b) channel utilization, where E is the packet error rate, and G is the traffic load in each mobile (channel rate: 10 Mbps).

with the delay and jitter bounds analyzed in Section VI. Both voice and video traffic have the same trends for delay and jitter which increase as the number of mobiles or the traffic load increases. Receiver playback buffers can be used to smooth the jitter [48], [49].

VIII. Conclusion

This paper describes a scheduling algorithm for EC-MAC, a low-power access protocol for wireless and mobile ATM networks. The goals of the access protocol are to conserve battery power, to support multiple traffic classes, and to provide different levels of service quality for bandwidth allocation. The proposed algorithm is a *priority round robin with dynamic reservation update and error compensation* scheduling. Performance analysis is presented. Delay and jitter bounds, and rate guarantees for CBR and VBR traffic are provided. The simulation emphasizes on various quality-of-service parameters with varying number of mobiles in a cell. Currently, low-power operation is done by contiguous bandwidth allocation and by cooperating with EC-MAC. Future work will include more power adaptation by prioritizing low-power and high-lower mobiles in scheduling.

References

[1] K. M. Sivalingam, J.-C. Chen, P. Agrawal, and M. B. Srivastava, “Design and analysis of low-power access protocols for wireless and mobile ATM networks,” *ACM/Baltzer Mobile Networks Appl.,* 1999.

[2] ITU-T Rec. H.263, *Video Coding for Low Bit Rate Communication,* Mar. 1996.

[3] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, “On the self-similar nature of Ethernet traffic (extended version),” *IEEE/ACM Trans. Networking,* vol. 2, pp. 1–15, Feb. 1994.

[4] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, “Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level,” *IEEE/ACM Trans. Networking,* vol. 5, pp. 71–86, Feb. 1997.

[5] H. Zhang, “Service disciplines for guaranteed performance service in packet-switching networks,” *Proc. IEEE,* vol. 83, Oct. 1995.

[6] A. Varma and D. Stiliadis, “Hardware implementation of fair queuing algorithms for asynchronous transfer mode networks,” *IEEE Commun. Mag.,* vol. 35, pp. 54–68, Dec. 1997.

[7] J.-C. Chen, K. M. Sivalingam, and P. Agrawal, “Performance comparison of battery power consumption in wireless multiple access protocols,” *ACM/Baltzer Wireless Networks,* 1999.

[8] L. Zhang, “Virtual clock: A new traffic control algorithm for packet switching networks,” *ACM Trans. Comput. Syst.,* vol. 9, pp. 101–124, May 1991.

[9] S. Golestani, “A self-clocked fair queueing scheme for broadband applications,” in *Proc. IEEE Infocom,* Toronto, Ont., Canada, June 1994, pp. 636–646.

[10] M. J. Karol, Z. Liu, and K. Y. Eng, "An efficient demand-assignment multiple access protocol for wireless packet (ATM) networks," *ACM/Baltzer Wireless Networks,* vol. 1, no. 3, pp. 267–279, 1995.

[11] D. Raychaudhuri and N. D. Wilson, "ATM-based transport architecture for multi-services wireless personal communication networks," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 1401–1414, Oct. 1994.

[12] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," in *Proc. ACM Sigcomm,* Palais des Festivals, Cannes, France, Sept. 1997.

[13] T. S. E. Ng, I. Stoica, and H. Zhang, "Packet fair queuing algorithms for wireless networks with location-dependent errors," in *Proc. IEEE Infocom,* San Francisco, CA, Apr. 1998, pp. 1103–1111.

[14] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architectures,* 2nd ed. Norwell, MA: Kluwer, 1997.

[15] S. Okubo, S. Dunstan, G. Morrison, M. Nilsson, H. Radha, D. L. Skran, and G. Thom, "ITU-T standardization of audiovisual communication system in ATM and LAN environments," *IEEE J. Select. Areas Commun.,* vol. 15, pp. 965–982, Aug. 1997.

[16] D. J. Goodman and S. X. Wei, "Efficiency of packet reservation multiple access," *IEEE Trans. Veh. Technol.,* vol. 40, pp. 170–176, Feb. 1991.

[17] P. Agrawal, E. Hyden, P. Krzyzanowski, P. Mishra, M. B. Srivastava, and J. A. Trotter, "SWAN: A mobile multimedia wireless network," *IEEE Personal Commun.,* vol. 3, pp. 18–33, Apr. 1996

[18] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in *Proc. ACM Sigcomm,* Sept. 1989, pp. 3–12.

[19] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Trans. Networking,* vol. 1, pp. 344–357, June 1993.

[20] P. Goyal, H. M. Vin, and H. Cheng, "Start-time fair queuing: A scheduling algorithm for integrated services packet switching networks," in *Proc. ACM Sigcomm,* Stanford, CA, Aug. 1996.

[21] J. C. R. Bennett and H. Zhang, "WF^{2}Q: Worst-case fair weighted fair queueing," in *Proc. IEEE Infocom,* San Francisco, CA, Apr. 1996, pp. 120–128.

[22] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round robin," in *Proc. ACM Sigcomm,* Sept. 1995, pp. 231–242.

[23] D. Stiliadis and A. Varma, "Frame-based fair queueing: A new traffic scheduling algorithm for packet-switched networks," in *Proc. ACM Sigmetrics,* May 1996.

[24] S. Suri, G. Varghese, and G. Chandranmenon, "Leap forward virtual clock: A new fair queuing scheme with guaranteed delays and throughput fairness," in *Proc. IEEE Infocom,* Kobe, Japan, Apr. 1997, pp. 558–566.

[25] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," *IEEE/ACM Trans. Networking,* vol. 3, pp. 365–386, Aug. 1995.

[26] J. C. R. Bennett and H. Zhang, "Hierarchical packet fair queueing algorithms," *IEEE/ACM Trans. Networking,* vol. 5, Oct. 1997.

[27] I. Stoica, H. Zhang, and T. S. E. Ng, "A hierarchical fair service curve algorithm for link-sharing, real-time and priority services," in *Proc. ACM Sigcomm,* Palais des Festivals, Cannes, France, Sept. 1997.

[28] D. Ferrari and D. Verma, "A scheme for real-time channel establishment in wide-area networks," *IEEE J. Select. Areas Commun.,* vol. 8, pp. 368–378, Apr. 1990.

[29] M. Vishnu and J. W. Mark, "HOL-EDD: A flexible service scheduling scheme for ATM networks," in *Proc. IEEE Infocom,* San Francisco, CA, Apr. 1996, pp. 647–654.

[30] D. Ferrari, H. Zhang, and D. Verma, "Guaranteeing delay jitter bounds in packet switching networks," in *Proc. TriComm,* Chapel Hill, NC, Apr. 1991, pp. 35–46.

[31] S. Golestani, "A stop-and-go queueing framework for congestion management," in *Proc. ACM Sigcomm,* Philadelphia, PA, Sept. 1990, pp. 8–18.

[32] C. Kalmanek, H. Kanakia, and S. Keshav, "Rate controlled servers for very high-speed networks," in *Proc. IEEE Globecom,* San Diego, CA, Dec. 1990, pp. 300.3.1–300.3.9.

[33] H. Zhang and D. Ferrari, "Rate-controlled static-priority queueing," in *Proc. IEEE Infocom,* San Francisco, CA, Apr. 1993, pp. 227–236.

[34] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. K. Tripathi, "Enhancing throughput over wireless LAN's using channel state dependent packet scheduling," in *Proc. IEEE Infocom,* San Francisco, CA, Apr. 1996, pp. 1133–1140.

[35] C. Fragouli, V. Sivaraman, and M. Srivastava, "Controlled multimedia wireless link sharing via enhanced class-based queuing with channel-state-dependent packet scheduling," in *Proc. IEEE Infocom,* San Francisco, CA, Apr. 1998, pp. 572–580.

[36] N. R. Figueira and J. Pasquale, "Remote-queueing multiple access (RQMA): Providing quality of service for wireless communications," in *Proc. IEEE Infocom,* San Francisco, CA, Apr. 1998, pp. 307–314.

[37] R. Kautz and A. Leon-Garcia, "A distributed self-clocked fair queueing architecture for wireless ATM networks," in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun. (PIMRC),* Helsinki, Finland, 1997, pp. 189–193.

[38] M. A. Arad and A. Leon-Garcia, "A generalized processor sharing approach to time scheduling in hybrid CDMA/TDMA," in *Proc. IEEE Infocom,* San Francisco, CA, Apr. 1998, pp. 1164–1171.

[39] S. Kishore, J.-C. Chen, K. M. Sivalingam, and P. Agrawal, "A battery power level aware MAC protocol for CDMA wireless networks," in *Proc. IEEE Int. Conf. Universal Personal Commun. (ICUPC),* 1998.

[40] D. Saha, S. Mukherjee, and S. K. Tripathi, "Multirate scheduling of VBR video traffic in ATM networks," *IEEE J. Select. Areas Commun.,* vol. 15, pp. 1132–1147, Aug. 1997.

[41] J. R. Jump, *YACSIM Reference Manual.* Rice Univ., Dept. Elect. Comput. Eng., 2.1 ed., Mar. 1993.

[42] "Digital video coding at Telenor R&D," [online] Available http://www.fou.telenor.no/brukere/DVC/.

[43] M. Zorzi, R. Rao, and L. B. Milstein, "On the accuracy of a first-order Markov model for data transmission on fading channels," in *Proc. IEEE Int. Conf. Universal Personal Commun. (ICUPC),* Nov. 1995, pp. 211–215.

[44] A. Chockalingam, M. Zorzi, and R. Rao, "Performance of TCP on wireless fading links with memory," in *Proc. IEEE Int. Conf. Commun. (ICC),* Atlanta, GA, June 1998.

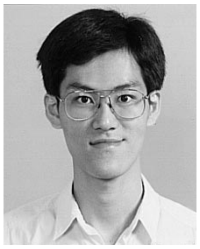
[45] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurthi, "Packet reservation multiple access for local wireless communications," *IEEE Trans. Commun.,* vol. 37, pp. 885–890, Aug. 1989.

[46] R. Aravind, M. R. Civanlar, and A. R. Reibman, "Packet loss resilience of MPEG-2 scalable video coding algorithms," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 6, pp. 426–435, Oct. 1996.

[47] J. Zhang, M. R. Frater, J. F. Arnold, and T. M. Percival, "MPEG 2 video services for wireless ATM networks," *IEEE J. Select. Areas Commun.,* vol. 15, pp. 119–128, Jan. 1997.

[48] N. Shivakumar, C. J. Sreenan, B. Narendran, and P. Agrawal, "The Concord algorithm for synchronization of networked multimedia streams," in *Proc. IEEE Int. Conf. Multimedia Comput. Syst.,* May 1995, pp. 31–40.

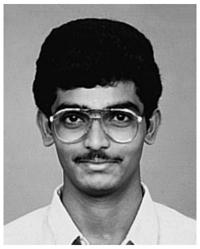
[49] P. Agrawal, J.-C. Chen, and C. J. Sreenan, "Use of statistical methods to reduce delays for media playback buffering," in *Proc. IEEE Int. Conf. Multimedia Comput. Syst.,* Austin, TX, June 1998, pp. 259–263.



Jyh-Cheng Chen (S'96–M'99) received the B.S. degree in information science from Tunghai University, Taichung, Taiwan, R.O.C., in 1990. In 1992, he received the M.S. degree in computer engineering from Syracuse University, Syracuse, NY. He received the Ph.D. degree in electrical engineering from the State University of New York, Buffalo in 1998.

From 1995 to 1996, he worked as a software engineer at ASOMA-TCI, Inc., North Tonawanda, NY. During the summer of 1997, he worked in the Networked Computing Technology Department at AT&T Labs, Whippany, NJ, and contributed to energy efficient MAC protocols for wireless ATM networks. Since August 1998, he has been a Research Scientist in applied research at Telcordia Technologies (formerly Bellcore), Morristown, NJ. At Telcordia Technologies, his research is focused on third-generation wireless networking, next generation networks, QoS for IP networks, Internet telephony, and active networks.

Dr. Chen is a member of ACM.



Krishna M. Sivalingam (S'92–M'95) received the B.E. degree in computer science and engineering in 1988 from Anna University, Madras, India. He received M.S. and Ph.D. degrees in computer science from the State University of New York (SUNY), Buffalo, in 1990 and 1994, respectively.

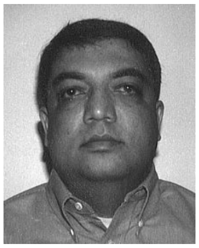
While at SUNY Buffalo, he was a Presidential Fellow from 1988 to 1991. He is currently an Assistant Professor in the School of Electrical Engineering and Computer Science at Washington State University, Pullman. Prior to that, he was an Assistant Professor at the University of North Carolina Greensboro from 1994 until 1997. He has conducted research at Lucent Technologies' Bell Labs, Murray Hill, NJ, and at AT&T Labs, Whippany, NJ, where he also served as a Consultant during 1997. His research interests include wireless computing and communications, all-optical WDM networks, ATM networks, high-speed communication networks such as Gigabit Ethernet, high-performance distributed computing, and performance evaluation.

Dr. Sivalingam has served on the Conference Committees of IEEE INFOCOM 1997, the WOWMOM Workshop 1998 and 1999, ACM Mobicom 1999, MASCOTS 1999, and Mobile Data Access 1999. He is a member of ACM.

Prathima Agrawal (S'74–M'77–SM'86–F'89) received the B.E. and M.E. degrees in electrical communication engineering from the Indian Institute of Science, Bangalore, India, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles.

She is a Chief Scientist and Assistant Vice President of the Internet Architecture Research Laboratory at Telcordia Technologies (formerly Bellcore), Morristown, NJ. From 1997 to 1998, she was head of the Networked Computing Technology Department at AT&T Labs, Whippany, NJ. Prior to that, she headed the Networked Computing Research Department at Bell Labs, Murray Hill, NJ. Her research interests are computer networks, mobile computing, parallel processing, and VLSI CAD. She has published more than 150 papers and has received or applied for more than 30 US patents.

Dr. Agrawal is a Member of the ACM. Presently, she chairs the IEEE Fellow Selection Committee.



Raj Acharya received the Ph.D. degree from the University of Minnesota/Mayo Graduate School of Medicine in 1984.

He worked with the Dynamic Spatial Reconstructor Project at the Mayo Clinic from 1981 to 1985. From 1984 to 1986, he was a Research Scientist at GE-CGR, Paris, France. He is currently a Professor of Computer Science and Engineering at the State University of New York (SUNY), Buffalo. His research interests are in multimedia networks, multimedia information retrieval, fractals, image processing, and visualization.

Dr. Acharya is on the Editorial Board of the *Journal of Computerized Medical Imaging and Graphics*. He was General Chair of the SPIE Conference on Biomedical Image Processing in 1992 and 1993. He was General Chair of the 1994 SPIE International Conference on Physiology and Function from Multidimensional Images. He was also cochair of the 1994 IEEE Workshop in Biomedical Image Analysis. He was on the Scientific Board of the IEEE Model Based 3-D Biomedical Image Analysis. His research work has been featured in *Businessweek*, 1996 Mathematics Calendar, *The Scientist, Diagnostic Imaging,* and *Biomedical Engineering Newsletter*.