A Slotted CDMA Protocol with BER Scheduling for Wireless Multimedia Networks

Ian F. Akyildiz, *Fellow, IEEE,* David A. Levine, and Inwhee Joe

***Abstract*—In future wireless multimedia networks, there will be a mixture of different traffic classes which have their own maximum tolerable bit error rate (BER) requirements. In this paper, a novel medium access control (MAC) protocol called wireless multimedia access control protocol with BER scheduling (in short form, WISPER) for CDMA-based systems is proposed. WISPER utilizes the novel idea of scheduling the transmission of multimedia packets according to their BER requirements. The scheduler assigns priorities to the packets, and performs an iterative procedure to determine a good accommodation of the highest-priority packets in the slots of a frame so that packets with equal or similar BER requirements are transmitted in the same slots. The proposed WISPER protocol has been validated using a software emulator on the cellular environment. Performance evaluation results based on the implementation are also included in the paper.**

***Index Terms*—BER scheduling, code division multiple access, multimedia traffic, power control, priority, wireless networks.**

I. Introduction

IT IS anticipated that traffic in next-generation cellular wireless networks will be a mixture of voice, data, and video messages. Most services and applications currently available in wireline networks will be adapted and extended to the wireless environment. Packets generated by future mobile platforms will belong to one of several traffic classes. Each of these classes will exhibit a large variety of characteristics and inherent requirements, such as transmission rate, maximum tolerable bit error rate (BER), and timeout specifications. Because of the expected variability in the transmitted traffic, it is predicted that traditional voice-based medium access control (MAC) protocols such as [7], [25], [29] would perform poorly in future wireless networks. What is required is a new generation of highly flexible MAC protocols that can easily adapt to the changing conditions and requirements of projected multimedia traffic over future wireless networks.

Code-division multiple access (CDMA) has emerged as one of the most promising multiple-access techniques for future wireless multimedia networks [18] and has been selected for IMT-2000 systems by standadization bodies all around the world [14]. For voice CDMA systems based on the IS-95 standard, power control is used to combat the near–far problem by maintaining nearly constant received power at the base station [6]. If this conventional power-control scheme applies for multimedia traffic without any modification, the capacity is largely limited by the traffic with the lowest BER requirement [1], [5], [6], [26]. In other words, voice packets can typically tolerate BER's of up to 10^{-3}, while data packets require BER's below 10^{-9}. Consequently, it is wasteful to schedule simultaneous transmissions of voice and data packets, since the transmission channel must be able to satisfy the most stringent BER specification among all the packets that are being transmitted at the same time.

The conventional closed-loop power control can easily be extended to support multimedia traffic by assigning different power levels to different traffic types [23]. Likewise, this scheme maintains the received power at a fixed level for each traffic type no matter what the transmission rate is. More recently, several approaches have been released to propose optimal power control for multimedia multirate traffic in the sense of maximizing the capacity or minimizing the total transmit power [28], [27], [24], [10]. In [27] and [24], the optimal power distribution is obtained through the use of channel measurement information.

Policies which schedule the order of transmissions for multimedia packets will have a great impact on the efficiency and performance of MAC protocols for future wireless networks. The design of an efficient "packet scheduler" is a difficult task that typically involves a large number of conflicting requirements which must be analyzed and weighted before a balanced and "fair" solution can be found. Several criteria can be used as guidelines for the design of an efficient packet scheduler, e.g., maximization of throughput, minimization of packet losses, upholding of quantitative quality of service (QoS) guarantees, and scheduling according to a pre-defined priority structure. Several scheduling disciplines have been proposed for guaranteed performance service in wireline packet-switching networks [30]. More recently, an interesting algorithm has been introduced for time-division multiple access (TDMA)-based networks to schedule transmissions based on different priority levels assigned to the different asynchronous transfer mode (ATM) service classes (i.e., the highest priority for constant bit rate (CBR) services), and on the delay constraints of each active connection [15]. For CDMA-based systems, new methods appeared in [16], [19], and [3]. While [16] and [19] are based on a CDMA scheme without a TDMA

Manuscript received June 6, 1996; revised October 10, 1998; approved by IEEE/ACM Transactions on Networking Editor K. Sabnani. The work of I. F. Akyildiz was supported in part by Korea Telecom. The work of D. A. Levine was supported in part by the National Science Foundation under Grant EID-9018632.

I. F. Akyildiz is with the Broadband and Wireless Networking Laboratory at School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: ian@ece.gatech.edu).

D. A. Levine is with BellSouth, Atlanta, GA 30385 USA (e-mail: dlevine@snt.bellsouth.com).

I. Joe is with the Network Research Group, Oak Ridge National Laboratory, Oak Ridge, TN 37831 USA (e-mail: inwhee@nrg.cind.ornl.gov).

Publisher Item Identifier S 1063-6692(99)03525-6.

1063–6692/99$10.00 © 1999 IEEE

component, a hybrid multiple-access scheme is used in [3], similar to our protocol introduced in this paper. However, only traffic rate and delay constraints are considered in [3] to assign resources and users with different BER requirements will be spread over the same interval.

The objective of these scheduling disciplines is to provide performance guarantees in terms of delay, delay jitter, throughput, and loss rate. These scheduling disciplines are implemented in real time, thus requiring algorithms of moderate computation complexity. A capacity allocation algorithm for wireless ATM networks, based on a transmission request/permit paradigm, is proposed in [21]. In this algorithm, the peak bit rate declared by each mobile terminal in the network is enforced by spacing transmission permits accordingly. All transmission requests are stored and serviced according to a first come first served (FCFS) discipline. A demand-assignment MAC protocol based on multicode (MC) direct-sequence (DS) CDMA was recently proposed in [13]. In this protocol, the transmission scheduling policy takes into account both the maximum transmission rate capabilities of a mobile terminal, as well as the transmission rate being requested. Note that the packet timeouts are ignored in [13] and [21].

In this paper, we introduce the WISPER protocol, a novel MAC protocol for CDMA systems that tailors the transmission of multimedia packets according to their BER requirements. WISPER can be classified as a slotted and demand-based assignment protocol. Considering that wireless bandwidth is perhaps the most scarce and precious resource in a wireless communication system, the main objectives of the WISPER protocol are to maximize the throughput and to minimize the packet losses. WISPER contains a novel work-conserving service discipline for scheduling the packet transmissions. In WISPER, the throughput is maximized by ordering packet transmissions according to traffic classes and by scheduling the packet transmissions at the mobile terminal's maximum possible transmission rate. In addition, packet losses are minimized by using a novel packet-prioritization scheme that determines packet transmission order by the remaining time until the packet timeout. Since our protocol classifies packets into different slots based on their BER target, the conventional power-control scheme can be used with one power level for each slot, which is simple to implement compared to other power control methods using different power levels at the same time for each slot.

The remainder of this paper is organized as follows. In Section II, we provide an overview of the proposed protocol. The overview includes a description of the CDMA system model being considered, as well as a description of the protocol operation. In Section III, we present a description of the novel packet scheduler developed for the WISPER protocol. We analyze the components of the scheduler mechanism and provide an example of the packet accommodation process. In Section IV, we present a description of the traffic models developed for the performance evaluation of the protocol. In Section V, we examine the performance of the protocol, state our assumptions, describe simulation models, and analyze the results. Finally, in Section VI we conclude the paper by highlighting our contribution.

II. Slotted CDMA with BER Scheduling Protocol

A. The CDMA System Model

We consider a wireless cellular network system that uses MC DS-CDMA (MC-CDMA) similar to the system model described in [2] and [12]. In this model, all data packets are transmitted at a "basic" rate. To increase its transmission rate over the basic rate, a mobile terminal may be able to transmit up to m packets simultaneously, thus becoming an m-rate mobile terminal. An active mobile terminal n admitted to the system is assigned a primary pseudonoise (PN) code C_n^{PN}. In order to transmit at rates higher than the basic rate, mobile terminal n uses different spreading codes $C_n^{(i)}$ $(i = 1, \cdots, m)$ for each basic-rate stream to be transmitted. The different spreading codes $C_n^{(i)}$ are derived from the primary PN code C_n^{PN} by

$$C_n^{(i)} = C_n^{\mathrm{PN}} \times D_i, \quad D_i \perp D_j, \qquad i \neq j.$$

where $D_i (D_i \perp D_j, i \neq j)$ are from a set of orthogonal codes (e.g., Walsh codes), so that $C_n^{(i)} \perp C_n^{(j)}$, $i \neq j$ is guaranteed. In theory and because of this orthogonality, multiple streams transmitted from the same mobile terminal do not interfere with each other. In reality, there is always some mutual interference. We assume that each mobile terminal that is accepted into a cell is assigned a different primary PN code. Primary codes can be "reused" after a mobile terminal finishes its wireless connection. As long as there are no conflicts, a mobile terminal will keep the same PN code as it travels through different cells in the network.

B. The Basic Protocol

We consider a wireless network that can support several classes κ_q, for $q = 1, 2, \cdots$, of multimedia traffic (e.g., voice, data, email, video, audio, fax, etc.). Each of these traffic classes presents specific characteristics such as constant or variable bit rate (VBR), activity factor, and timeouts, as well as specific QoS requirements, such as maximum tolerable BER's, packet error rates (PER), and packet loss rates (PLR). We assume that mobile terminals generate packets in batches, where all packets in a batch have the same timeout specification. In our protocol, the total available bandwidth is divided in two bands, one for the uplink, the other for the downlink. For both bands, time is divided into frames of length τ_f. The frame length is chosen so as to coincide with the packet arrival rate of the most abundant traffic class (usually voice). A typical frame length is in the order of 16 ms [8]. For the uplink, each frame is divided into N_p packet slots and one request slot, as shown in Fig. 1. Each packet slot can carry any class of traffic. The request slot can be used for two purposes: 1) to place admission requests by new mobile terminals that want to be admitted to the wireless network and 2) to place transmission requests by mobile terminals currently registered in the wireless network.

When a mobile terminal wants to be admitted to the wireless network, it selects at random a PN code from a pool of codes

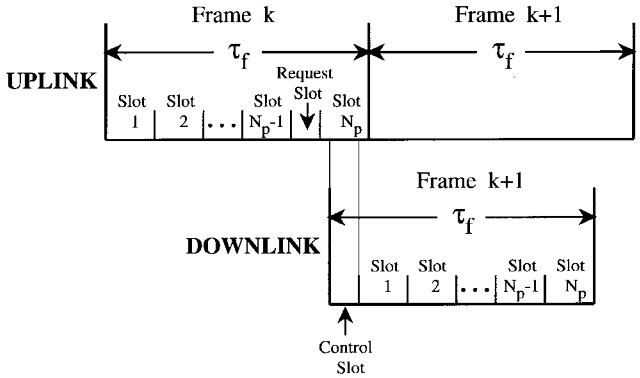


Fig. 1. Uplink and downlink channels: timing diagram.

specifically reserved for this purpose. Using this code, the mobile terminal places an admission-request packet in the request slot. The request slot contains the mobile terminal identification number, information about the traffic class to be transmitted, the mobile terminal's maximum transmission rate capabilities, and, if applicable, the bit rate characteristics (such as lower and maximum bit rate limits) of the traffic to be transmitted. Collisions can occur in the request slot if there are two or more admission requests using the same PN code, or if the total number of simultaneous request packets results in unmanageable mutual interference. If no collisions occur in the request slot, the packet was received successfully, and if the call admission controller (CAC) determines that the QoS requirements of other mobile terminals can still be satisfied after this mobile user is admitted, the mobile terminal receives an acceptance notification. The acceptance information includes a unique identification number and a primary PN code for the exclusive use of the mobile throughout its connection lifetime.

Whenever a mobile terminal has new packets ready for transmission, it must send a transmission request to the base station indicating the number of packets in the new batch, as well as the corresponding timeout value of the packets (conversely, the mobile terminal can specify the generation time of the packets, so that the timeout value can be deduced). A mobile terminal sends the transmission request either by using the request slot or by piggybacking the request in a previously transmitted data packet. The latter method is used whenever possible in order to reduce contention in the request slot. In either case, requests are transmitted using the mobile terminal's assigned primary PN code. Once a request has been received, a data structure is used by the base station to keep track of the batch associated with the request. The data structure contains information such as the mobile terminal that owns the batch, the packets' timeout value, and the number of packets in the batch. This information is kept until the packets in the batch have been received successfully, or until they timeout and are discarded.

For the downlink, the base station has sole control of the downstream traffic, and thus, request slots for the base station are not needed. As shown in Fig. 1, downstream frames are also of length τ_f, and are divided into N_p packet slots and a control slot. The control slot is used by the base station to provide acceptance notification to mobile terminals that requested admission to the network, and to provide transmission instructions to mobile terminals that previously requested permission to transmit packets. A base station does not need to send an identification number when it tries to communicate with a given mobile terminal. Rather, the base station simply needs to transmit its message using the same primary PN code that is being used by the mobile terminal. When the base station responds to a request for packet transmission, it specifies the slot(s) and the corresponding number of packets that can be transmitted in the next frame. The control slot can also be used by the base station to provide acknowledgment information on the packets that were successfully received in a previous frame.

Fig. 1 also shows the relative timing of the upstream and downstream frames. For the upstream frame, the request slot cannot be the last slot in the frame, since the base station needs time to process the requests and to compute the slot assignments before the next upstream frame begins. The exact position of the request slot will depend on the processing capabilities at the base station. Also, since the mobile terminals need to have their transmission slot assignments on time, the downstream frame is not aligned with the upstream frame and always begins with the control slot. In addition, note that the end of the control slot does not coincide with the beginning of the next upstream frame. This is so that mobile terminals have time to process the base station's replies. Note also that the size of the control and request slots is not necessarily equal to the size of the data slots (typically, control slots will be smaller than data slots).

The WISPER protocol designates slots that can support certain BER's, and it schedules packet transmissions in these slots in such a way that the wireless bandwidth can be used efficiently. Consider that each traffic class κ_q, for $q = 1, 2, \cdots$, has a maximum BER specification given by $B(\kappa_q)$. The BER in a communication channel is a function of several factors, e.g., thermal noise, interference, received signal power, modulation, and forward error-correction scheme being employed, etc. Obviously, a factor of fundamental importance in the determination of the BER is the number of mobile terminals that are concurrently transmitting in the same channel bandwidth. This is because in CDMA, from the perspective of a transmitter, any other simultaneous transmission in the same frequency band contributes as noise to the transmitted signal. We assume that mobile terminals do not need to send continuous data to keep the base station in synchronization. Also, we assume that the maximum number of simultaneous transmissions which can be allowed without exceeding a certain BER can be determined for the current system. In other words, we assume that if the maximum tolerable BER values b_q, for $q = 1, 2, \cdots$, are given *a priori*, we can always determine the maximum number of simultaneous transmissions $S(b_q)$, for $q = 1, 2, \cdots$, that can be allowed so that the corresponding BER values are not exceeded. The maximum tolerable BER values b_q are the BER values of the *transmission channel*. Obviously, the transmission BER values can be decreased by using adequate error-correction coding techniques.

To use the available bandwidth in an efficient manner, packets that have either equal or almost the same maximum

BER specifications should be transmitted in the same slot. Obviously, if two packets h and k with maximum tolerable BER specifications $b_h > b_k$ are scheduled for transmission in the same slot, then the maximum number of simultaneous transmissions in this slot will be limited by $S(b_k)$. An efficient organization of the packets within the slots of a frame is not limited to grouping packets with the same or similar maximum tolerable BER specifications in the same slot. If, at a given time, there are more outstanding packets than those that can be transmitted in the next frame, then it becomes necessary to select those packets that should be transmitted first (in the next frame). The selection of these packets can be made according to several criteria, such as packet timeout deadline proximity, packet priority, and other considerations, so that certain performance bounds (e.g., maximum delay and delay jitter, packet loss rate) are not violated. In the next section, we proceed to describe a novel scheduler that prioritizes and organizes the transmission of packets by taking into consideration some of these criteria.

III. Packet Scheduler

The design of an efficient packet scheduler for our protocol is a difficult task which is heavily dependent on a large number of factors, e.g., available bandwidth, number of users, characteristics and requirements of the different traffic classes, and QoS requirements of the transmitted traffic. Indeed, it is impossible to design an "optimal scheduler" that is "fair" for every mobile terminal in the network, because the design typically includes a large number of conflicting requirements that must be weighted before a solution can be implemented. Thus, the emphasis should be to design an efficient scheduler that satisfies as many requirements as possible at the same time.

Considering the fact that the available bandwidth is perhaps the most precious resource in a wireless communication system, the main objectives of our scheduler are to maximize the throughput (given a certain traffic-class priority structure) and to minimize packet losses. These objectives are achieved by a scheduler that performs two tasks: 1) determination of packet priorities and 2) determination of the position (slot within the next frame) where a packet should be transmitted so that the total frame throughput is maximized. These tasks are performed by the *packet prioritizer* function and the *packet usher* procedure, respectively.

The packet priority value is used primarily when there are more packets ready for transmission than those that can be accommodated in the next frame. When this is the case, only those packets with the highest priority are selected for immediate transmission. The computation of packet priorities is done dynamically, i.e., at the end of each frame. The assigned priority value is a function of the remaining time before the packet times out (when it is no longer useful and is discarded). The packet usher procedure selects the packets that are candidates for transmission in the next frame, and calls the *packet allocator* procedure to determine the exact positions in the next frame where the packets can be transmitted. The actual slot position within the frame is a function of the number of packets from the same and other traffic classes that need to be transmitted in the same frame. Several packets with different priorities can be scheduled for transmission in the same slot, as long as the packets have equal or similar BER requirements.

Next, we discuss in detail how the priority of a packet is determined, and the procedures that are followed to determine an appropriate allocation of packets within a frame so that the throughput is maximized.

A. Priority Determination

As explained in Section II, when a mobile terminal has new packets ready for transmission, it has to send a request to the base station, indicating the number of packets in the new batch as well as their timeout value. Upon receiving this request, the base station uses the packet prioritizer function to compute a transmission priority value for the new packets. This value is used to determine if the packets can be transmitted in the next frame. The packet prioritizer function does not determine the priority of individual packets, but rather the priority of the batch to which the new packets belong. Thus, at a given time, all packets within a batch have identical priority values, although not all packets within a batch may be sent in the same frame.

Since one of the objectives of the WISPER protocol is to minimize packet losses, the priority of a batch in the WISPER protocol is inversely proportional to the number of frames left before the packets in the batch timeout. Also, in order to give preference to a batch with a large number of packets over another batch with fewer packets, in the WISPER protocol the priority of a batch is directly proportional to the number of packets left in the batch. The packet prioritizer function computes for each batch a priority value that reflects the maximum number of slots that a mobile terminal should be allocated in each frame, assuming that the mobile terminal will transmit in slots that are evenly spaced throughout the remaining frames before its packets time out. The priority computation also assumes that each mobile terminal will transmit packets at its maximum transmission rate whenever possible. The priority value is used to limit the number of packets that a mobile terminal can send when its present cell is heavily loaded. Obviously, under light load conditions, a mobile terminal may be able to transmit all of its packets in a single frame.

Let M_n denote the maximum transmission rate (maximum number of packets that can be transmitted simultaneously in a slot) for mobile terminal n. Let $F_\beta(t)$ represent the number of frames left at the present time t before timeout of the packets in batch β, and let $P_\beta(t)$ represent the number of packets left in batch β, also at the present time t. Since the base station receives transmission requests and grants transmission permissions, the base station knows how many packets are ready for transmission in each mobile terminal. Also, since the base station receives the timeout values for the packets in a transmission request, the base station can determine when the packets in a batch β time out, i.e., when $F_\beta(t) = 0$. When this happens, the base station assumes that the packets will be discarded, and updates its records accordingly. The priority

$\Phi_\beta(t)$ of the packets in batch β at time t is determined with the packet prioritizer function

$$\Phi_\beta(t) = \begin{cases} \dfrac{\lceil P_\beta(t)/M_n \rceil}{F_\beta(t)}, & \text{if } \dfrac{\lceil P_\beta(t)/M_n \rceil}{F_\beta(t)} \le N_p \\ N_p, & \text{otherwise} \end{cases} \tag{1}$$

where N_p represents the number of slots in a frame.

From (1), it is clear that the priority of batch is directly proportional to the minimum number of slots required to send the packets in a batch at the maximum transmission rate of the mobile terminal, and inversely proportional to the number of frames left before the packets time out. That is, the priority is determined by two factors: minimum number of slots required and number of frames before timeout. Since the relative effects of the two factors on the priority are almost the same in terms of packet losses, the same first order can be used for both factors as in the expression above. Also from (1), we observe that the range of values for the priority of a batch β is $0 < \Phi_\beta(t) \le N_p$. As is explained next, the priority value of a batch is used by the packet usher procedure to determine if a mobile terminal should be granted transmission privileges for the next frame, and if so, the number of packets that will be allowed to send.

B. Computation of the Number of Packets to be Transmitted per Batch

The objective of the packet usher procedure is to accommodate packets from the batches that have the highest priority in such a way that the throughput in the next frame is maximized. The priority of a batch reported by the packet prioritizer function implicitly contains information on the maximum number of packets that the packet usher procedure should allow a mobile to send under high bandwidth demand conditions. This number is determined as follows.

Consider a mobile terminal n with a maximum transmission rate M_n that has a batch β with $P_\beta(t)$ packets waiting for transmission at time t. If the batch has been selected for transmission in the current frame, and if the priority of the batch for the next frame is $\Phi_\beta(t)$, then the number of packets $N(\beta)$ that mobile terminal n will be allowed to transmit during the next frame is determined according to

$$N(\beta) = \begin{cases} \lceil \Phi_\beta(t) \rceil \cdot M_n, & \text{if } \lceil \Phi_\beta(t) \rceil \cdot M_n < P_\beta(t) \\ P_\beta(t), & \text{otherwise} \end{cases} \tag{2}$$

Since the priority of a batch reflects the number of slots per frame that a mobile terminal should use for transmitting its packets, the actual number of packets that will be transmitted in the next frame by this mobile terminal is equal to the priority of the batch times the maximum transmission rate capability of the mobile terminal. In the presence of several contending transmitters, in MC-CDMA it is better that a transmitter sends several packets in the same slot rather than distributed in different slots. As explained in Section II-A, this is based on the fact that a mobile terminal with simultaneous transmissions uses derived PN codes that are orthogonal to each other, and thus, packets being transmitted simultaneously from the same mobile terminal practically do not interfere with each other.

```
Procedure Packet_Usher
begin
    for w = 1 to w ≤ q do begin
        update_priorities_in_queue(w);
    end;
    priority ← next_priority(∞);
    ptr ← 0;
    while( 1 ) begin
        for x = 1 to x ≤ q do begin
            if (ptr == 0) then do begin
                for y = 1 to y ≤ number_of_batches(κ_1) do begin
                if Priority(κ_1^y) ≥ priority then begin
                    to_send ← packet_calc(Priority(κ_1^y));
                    num_alloc ← Packet_Allocator(n, M_n, to_send, κ_1^y);
                end;
            end;

                .
                .
                .
            if (ptr = q − 1) then do begin
                for y = 1 to y ≤ number_of_batches(κ_q) do begin
                if Priority(κ_q^y) ≥ priority then begin
                    to_send ← packet_calc(Priority(κ_q^y));
                    num_alloc ← Packet_Allocator(n, M_n, to_send, κ_q^y);
                end;
            end;
        end;
        ptr ← ((ptr + 1) mod  q;

        if (num_alloc == 0) then return;

        if (priority ← next_priority(priority)) > 0 then continue
        else
            if there are batches left then begin
                for w = 1 to w ≤ q do begin
                    update_priorities_in_queue(w);
                end;
                priority ← next_priority(∞);
                continue;
            end
        end
        return
    end
end
```

Fig. 2. Packet usher procedure.

Thus, the packet usher procedure tries to schedule mobile terminals so that they transmit packets at their maximum transmission-rate capability.

1) The Packet Usher Procedure: The base station keeps up to q ordered data records of priority values. Each of these records contains the priority values (in decreasing order) of all packets waiting for transmission in mobile terminals which are under the control of the base station. There is one record for each of the different traffic classes κ_q, for $q = 1, 2, \cdots$. The different data records are searched for batches with packets that "qualify" for transmission. A batch has packets that qualify for transmission if the priority of the batch is equal to the value of a variable called "*priority*." When this occurs, the packet usher will try to accommodate packets from this batch in the next frame by using the packet allocator procedure (described below).

The data records are searched in a round-robin fashion. The search starts with the last data record that was searched the previous time the packet usher procedure was called. At the beginning of the packet usher procedure, the "*priority*" variable takes the value that corresponds to the highest batch priority among all data records. The value of the *priority* variable remains constant while the q data records are searched. Before initiating another search through the q data records, the variable *priority* takes a value that is equal to the priority of the highest priority batch among all surviving (i.e., nonempty) batches that remain in the system.

```
Procedure Packet_Allocator( n, M_n, P_n, κ_r )
begin
    for each (x ∈ N_p) do begin
        if slot x is full then continue;
        if slot x is empty then SLOT_CLASS[x] ← κ_r;
        find number of packets that can be placed in slot x;
        find number of packets P_x ≤ P_n that mobile n can transmit in slot x;
        P_n ← P_n − P_x;
        SLOT[x] ← SLOT[x] +P_x;
        NUM_PACKETS[x,n] ← P_x;
        if (P_n == 0) then return;
    end

    if (κ_r > κ_1) do begin
        for y = κ_{r−1} to y ≥ κ_1 step -1 do begin
            for each (x ∈ N_p) do begin
                if slot x is full then continue;
                if (SLOT_CLASS[x] == y) do begin
                    find number of packets that can be placed in slot x;
                    find number of packets P_x ≤ P_n that mobile n can transmit in slot x;
                    P_n ← P_n − P_x;
                    SLOT[x] ← SLOT[x] +P_x;
                    NUM_PACKETS[x,n] ← P_x;
                end
            end
        end
    end

    if (κ_r == κ_q) then return
    for y = κ_{r+1} to y ≤ κ_q do begin
        for each (x ∈ N_p) do begin
            if slot x is full then continue;
            if (SLOT_CLASS[x] == y) do begin
                if slot x cannot be converted to class κ then return
                SLOT_CLASS[x] ← κ_r;
                find number of packets that can be placed in slot x;
                find number of packets P_x ≤ P_n that mobile n can transmit in slot x;
                P_n ← P_n − P_x;
                SLOT[x] ← SLOT[x] +P_x;
                NUM_PACKETS[x,n] ← P_x;
            if (P_n == 0) then return;
            end
        end
    end
end
```

Fig. 3. Packet allocator procedure.

After all of the items in the q data records have been considered for transmission, all of the batches are "reprioritized", considering only the remaining packets in each batch. The search through the different data records continues until either there are no batches left, or until no more packets can be allocated in the next frame.

The pseudocode description for the packet usher procedure is presented in Fig. 2.

2) The Packet Allocator Procedure: The packet allocator procedure determines whether a number of packets can be accommodated in a frame, and keeps track of which slots contain which packets and of the maximum number of packets that can be accommodated in that slot. The packet allocator procedure requires four parameters:

1) mobile terminal identification number id;
2) maximum rate of transmission capability M_{id} of the mobile terminal;
3) number of packets P_{id} to be accommodated;
4) traffic class κ to be transmitted by mobile terminal id.

Depending on the traffic class of the packets to be accommodated, as well as on the slots that are still available in the next frame, the packet allocator procedure attempts the accommodation of packets according to the following criteria (and in the following order):

1) accommodation in empty slots or in slots that have packets with the same traffic class;
2) accommodation in slots that have packets with more stringent BER requirements;
3) accommodation in slots that have packets with more relaxed BER requirements.

The above criteria correspond to three stages in the packet allocator procedure. In the first stage, the allocation procedure searches for an empty slot in the frame, or for a slot that has the same traffic class κ_r. If such a slot is found, the allocator tries to accommodate all of the P_{id} packets in that slot. If not all of the packets can be accommodated, the allocator searches for other empty slots, of for other slots with the same traffic class. If the last slot is reached, and if there are still packets to be accommodated,

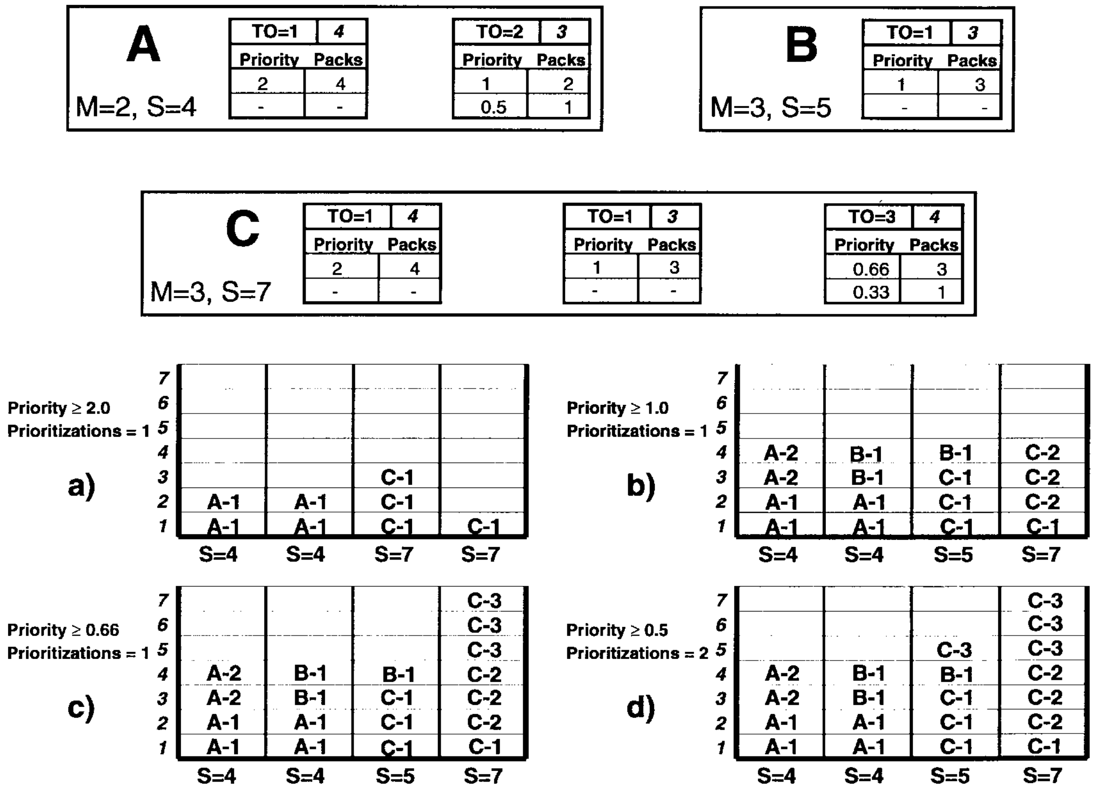


Fig. 4. Example of the packet accommodation procedure.

the packet allocator proceeds with the second stage of the algorithm.

For the second stage, the allocation procedure starts by searching all slots trying to find a slot with traffic class κ_{r-1}, where traffic class κ_{r-1} has more stringent BER requirements than traffic class κ_r. If such a slot is found, the allocator tries to place all of the remaining packets in the request. If there are still packets left to accommodate, and if all slots are either full or have packets from other traffic classes, the allocator tries to find slots that have a traffic class $\kappa_{r-2}, \cdots, \kappa_1$, until either all packets have been accommodated, or until slots with traffic class κ_1 have been considered. In other words, for the second stage of the procedure, the allocator tries to place packets inside slots that have more stringent BER requirements.

If there are additional packets that have not been accommodated, the allocator proceeds with stage three. The allocation procedure searches all slots trying to find a slot with traffic class κ_{r+1}. If such a slot is found, the allocator verifies that the slot can be converted to traffic class κ_r, i.e., the allocator verifies that the maximum number of packets required to support traffic class κ_r is not exceeded. Obviously, the allocator will convert the slot to traffic class κ_r only if there will be space left to accommodate at least one more packet. In a similar fashion, if the last slot is reached, and if there are packets left to be accommodated, the allocator searches for slots with traffic class $\kappa_{r+1}, \kappa_{r+2}, \cdots, \kappa_q$, and verifies if the slots can be converted to traffic class κ_r so that additional packets from the request can be accommodated.

Obviously, the packet allocator procedure ends when either all packets from a request have been accommodated, or until the end of stage three is reached. A pseudocode description for the packet allocator procedure is presented in Fig. 3.

3) Packet Allocation Example: To better understand the allocation algorithms described above, in Fig. 4 we present a simple example of the steps followed by a base station in order to achieve an efficient packet accommodation.

Consider the case where a base station knows that there are a total of 21 packets in six batches ready for transmission. The packets belong to one of three traffic classes. We denote these traffic classes as A, B, and C. Packets that belong to traffic class A have the most stringent BER requirement, which translates in that only up to $S = 4$ packets can be transmitted in a given slot if the maximum BER requirement for traffic class A is to be maintained in that slot. The corresponding maximum number of simultaneous transmissions for traffic classes B and C are $S = 5$ and $S = 7$. Also, we assume that all mobile terminals transmitting packets which belong to traffic class A have a maximum transmission capability (number of packets which can be transmitted at the same time) equal to $M = 2$. For mobile terminals transmitting packets of traffic classes B and C, the maximum transmission rate is $M = 3$.

For this example, we consider that the uplink frames are divided in frames with up to four slots. The base station maintains three priority records, since there are packets from three traffic classes. The state of the priority records, as well as snapshots of the allocation procedure, are also presented in Fig. 4, where each box within a priority record represents a batch of packets. The variable TO on the upper-left corner of each box is the timeout value, where $TO = 1$ implies that the batch is about to expire (one frame before timeout), and if it is not transmitted within the next frame, it will need to be discarded. The number on the upper right corner specifies the number of packets currently within the batch. The boxes under "Priority" and "Packs" are filled with the priority values which correspond to the total number of packets left in the batch, and the total number of packets which can be transmitted given that priority value. The priority value and the number of packets for the accommodation attempt are computed using (1)–(2).

At the beginning of the procedure, the highest priority value among all data records is determined (in this case the value is two). The packet usher procedure searches the different data records from left to right, top to bottom. For step a) in Fig. 4, only batches that have a priority equal to 2 are considered. The first batch that is found with this priority value is batch A-1. The four packets of this batch are assigned to slots 1 and 2, automatically establishing that the maximum number of packets that can be transmitted in these slot are $S = 4$. The procedure then finds that batch C-1 also has a priority equal to 2, searches for empty slots, and assigns the packets to slots 3 and 4, also establishing that the maximum number of packets that can be assigned to these slots is $S = 7$. No more batches can be found with a priority equal to 2. In step b), the next priority value (in this case 1) is used. Batch A-2 is found to have this priority, and thus, the procedure tries to accommodate two packets from this batch. The procedure searches for a slot with the same traffic class. Thus, the packets are placed in slot 1. Batch B-1 is also found to have a priority equal to 1. The procedure searches in vain for either an empty slot, or for a slot that has the same traffic class. Since no slot with those conditions can be found, the procedure searches for a slot that has a traffic class A. Slot number 1 is full, so the procedure places two packets in slot 2. But there is one more packet from batch B-1 that needs to be placed. The procedure now looks for a slot that has a traffic class C. Slot 3 is found, and before assigning the packet to this slot, the procedure verifies that the class of slot 3 can be upgraded to traffic class B. This means that slot 3 will now be allowed at most $S = 5$ packets. Finally, batch C-2 also has a priority equal to 1. The three packets from this batch are accommodated in the only slot that has a traffic class C, which in this case is slot 4. For step c), the next priority (0.666) is used. Only batch C-3 has packets with this priority. They are accommodated in slot 3. At this point, all of the batches have been processed. Since there is a slot that is not full (slot 3), the different batches are reprioritized considering only the remaining packets in each batch. The priority value is then set to 0.5. Only batch A-2 has a packet with this priority. Unfortunately, there is no slot that can support it. Thus, and since there is another batch (C-3) with a packet left, the next priority value is set to 0.33. Finally, in step d), the packet belonging to batch C-3 is placed in the only slot that can support this packet, in this case slot 3.

A random accommodation of the packets with highest priority could have resulted in a worst-case maximum capacity of only 16 packets in the frame. Since in this case a total of 20 packets were accommodated, the use of the packet-accommodation algorithm resulted in a net gain equal to 25% over the worst case.

TABLE I
NUMERICAL VALUES FOR THE SPEECH ACTIVITY MODEL

Condition	Average Duration (s)
Conversation Length	180.0
Principal Talkspurt	1.000
Principal Gap	1.350
Minispurt	0.275
Minigap	0.050

IV. TRAFFIC MODELS

To evaluate the performance of the protocol, we developed and simulated different traffic models:

- voice traffic (bursty traffic);
- CBR video traffic;
- VBR video traffic;
- CBR digital audio traffic;
- available bit rate (ABR) computer data traffic;
- ABR e-mail traffic.

The models generate six distinct traffic classes with notable differences among their characteristics and requirements, which makes them suitable to stress-test the performance of the protocol. The models also capture the most common traffic components in future wireless multimedia networks.

A. Models of Multimedia Traffic

Voice Traffic: This model is used to generate speech patterns in a conversation, and is based on the three-state Markov model presented in [8]. In this model, it is assumed that a speech source generates patterns of talkspurts and gaps. These patterns are the result of the talking, pausing, and listening behaviors in a conversation. Inside the talkspurts, there are also "minispurts" and "minigaps", the result of short activity and silent intervals that punctuate continuous speech.

The model assumes that in addition to the length of the conversations, all spurts and gaps have exponentially distributed durations, and that all durations of spurts and gaps are statistically independent of each other. We assume that mobile stations for voice traffic use vocoders that generate a data rate of 16.5 Kb/s during the minispurts periods of a conversation. Table I reports the numerical values used for the model.

CBR Video Traffic: In this model, a continuous bit stream is produced. The transmission time is assumed to be exponentially distributed, with a mean equal to 180.0 s. The constant bit rate is assumed to be equal to 220 kb/s.

VBR Video Traffic: This model, based on [4], attempts to mimic the bit rate characteristics of videophone and videoconference signals. In this multiple-state model, a state generates a continuous bit stream for a certain holding duration. The bit-rate values for the different states are obtained from a truncated exponential distribution. This distribution is defined with a minimum and a maximum bit-rate value. The states' holding times are assumed to be statistically independent and exponentially distributed. Call holding times are assumed to be exponentially distributed. Table II summarizes the numerical values used for the model.

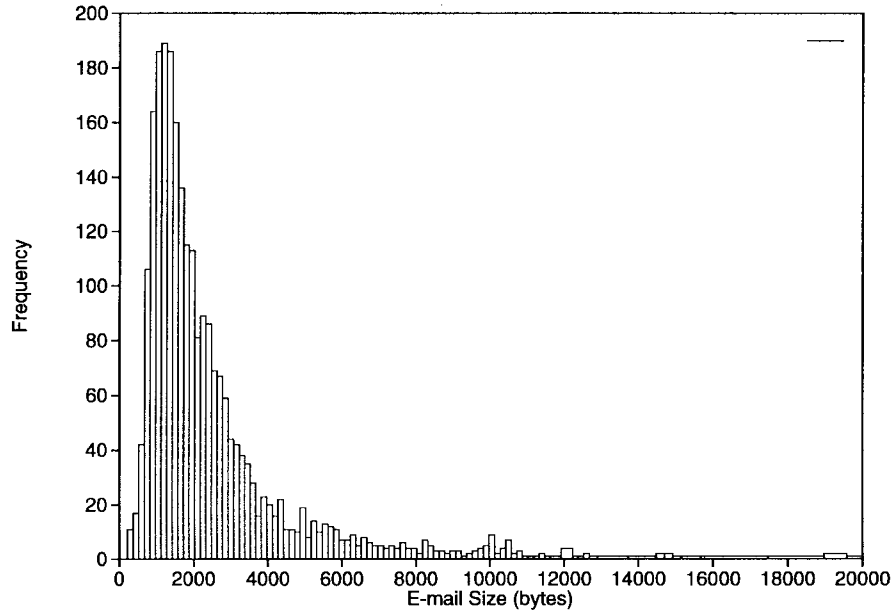


Fig. 5. Empirical size distribution of e-mail messages—sample histogram.

TABLE II
NUMERICAL VALUES FOR THE VBR VIDEO MODEL

Parameter	Value
Mean Video Transmission Time	180×10^3 msec
Mean State Holding Time	160 msec
Minimum Bit Rate	120 Kbps
Maximum Bit Rate	420 Kbps
Mean Bit Rate	239 Kbps

CBR Digital Audio Traffic: This model represents the production of a continuous bit stream of digital FM Stereo Audio. The parameters are a constant bit rate equal to 128 kb/s [20], and an audio call mean holding time of 360 s obtained from an exponential distribution.

ABR Computer Data Traffic: This is a simple model where the data message length is assumed to be exponentially distributed with a mean size equal to 30 kbytes.

ABR E-mail Traffic: We used an empirical distribution model for the generation of e-mail traffic. The empirical distribution was obtained after conducting an experiment where more than 2500 e-mail messages were analyzed. Fig. 5 presents the empirical size distribution used to simulate e-mail messages (only sizes less than 20 kbytes are shown). The mean e-mail size for this distribution is in the neighborhood of 2900 bytes.

TABLE III
SIMULATION ASSUMPTIONS AND SYSTEM PARAMETERS

Parameter		Value
Frame Size		16 msec
Number of Packet Slots per Frame N_p		10
Number of (Raw) Information Bits per Slot		264
Simulation Time (per point)		108,000 sec
Simulation Cycles (per point)		6.75×10^6
Voice Traffic	Maximum Tolerable BER	10^{-3}
	Maximum Number of Packets per Slot S	15
	Packet Time out Value	2 frames
	Maximum Transmission Rate Capability	1 packet/slot
CBR Digital Audio Traffic	Maximum Tolerable BER	10^{-4}
	Maximum Number of Packets per Slot S	10
	Packet Time out Value	6 frames
	Maximum Transmission Rate Capability	6 packets/slot
CBR Video Traffic	Maximum Tolerable BER	10^{-5}
	Maximum Number of Packets per Slot S	8
	Packet Time out Value	3 frames
	Maximum Transmission Rate Capability	4 packets/slot
VBR Video Traffic	Maximum Tolerable BER	10^{-6}
	Maximum Number of Packets per Slot S	6
	Packet Time out Value	3 frames
	Maximum Transmission Rate Capability	5 packets/slot
Computer Data Traffic	Maximum Tolerable BER	10^{-9}
	Maximum Number of Packets per Slot S	4
	Packet Time out Value	$\lceil 2 \times total\_packets \rceil$ frames
	Maximum Transmission Rate Capability	4 packets/slot
E-mail Traffic	Maximum Tolerable BER	10^{-9}
	Maximum Number of Packets per Slot S	4
	Packet Time out Value	$\lceil 50 \times total\_packets \rceil$ frames
	Maximum Transmission Rate Capability	1 packets/slot

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the WISPER protocol by simulation. Specifically, we evaluate the performance of the protocol in a single cell environment, considering seamless (always successful) handoffs. The modeling assumptions are the following.

- *Maximum Transmission Rates:* All mobile terminals transmitting a certain traffic class have the same maximum transmission rate (M) capability.
- *Generation of New Mobiles:* For each traffic class, a new mobile terminal can be generated in each frame, according to the Bernoulli process.
- *Transmission Requests and Transmission Instructions:* When an active mobile terminal generates new packets, it notifies the base station of this fact in the next possible opportunity (i.e., either by piggybacking the information at the end of a transmitted packet, or by using the next request slot). Transmission requests and transmission

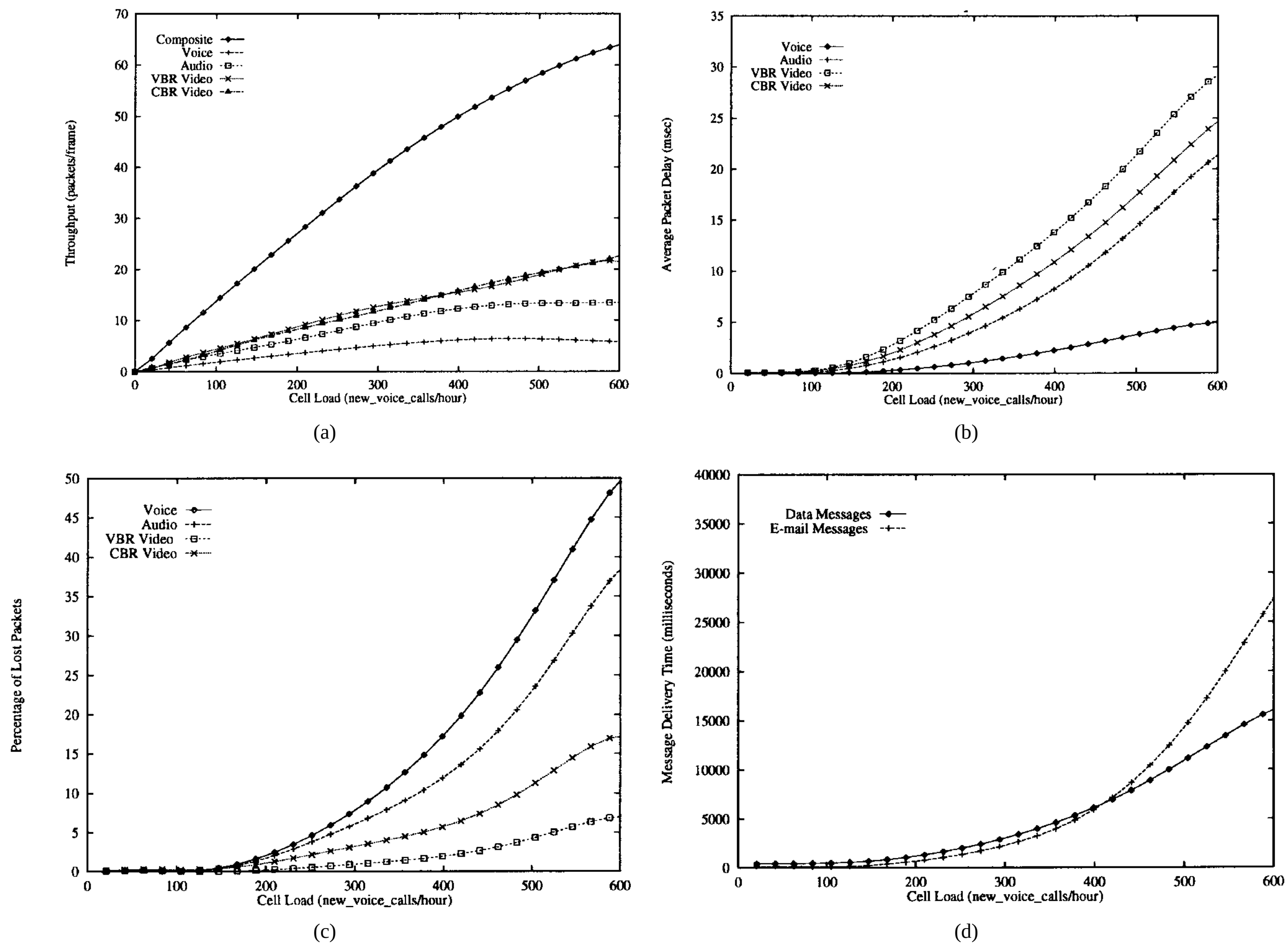


Fig. 6. Performance evaluation of the WISPER protocol. (a) Composite cell throughput. (b) Average packet delay. (c) Average packet loss. (d) ABR message delivery time.

instructions are always received successfully by the base station and the mobile terminals, respectively.

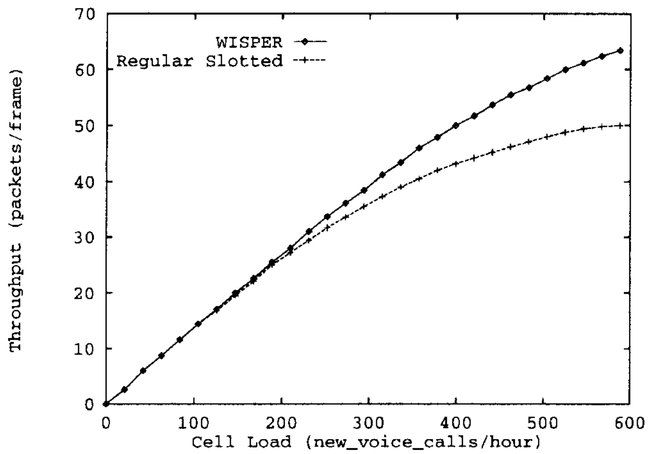
A list of the most important system parameters is given in Table III. The maximum number of packets that can be transmitted in each slot for each traffic class is determined based on the capacity expression for a CDMA system, presented in [6].

We evaluate the performance of the WISPER protocol under stress tests, i.e., when the cell load is progressively increased. The rates of new mobile terminal arrivals for different traffic classes are maintained constant throughout the simulations. The relative rates for new mobile terminal arrivals used in these experiments are given in Table IV. The main objective of the stress test experiments is to characterize the throughputs, average packet delays, packet loss ratios, and ABR traffic message delivery times under different traffic load values. The simulation results help us to capture the input load values that may result in unacceptable packet delays and packet losses for the traffic mix being considered. We also compare the performance of our protocol with a regular slotted-CDMA protocol using the conventional power control scheme.

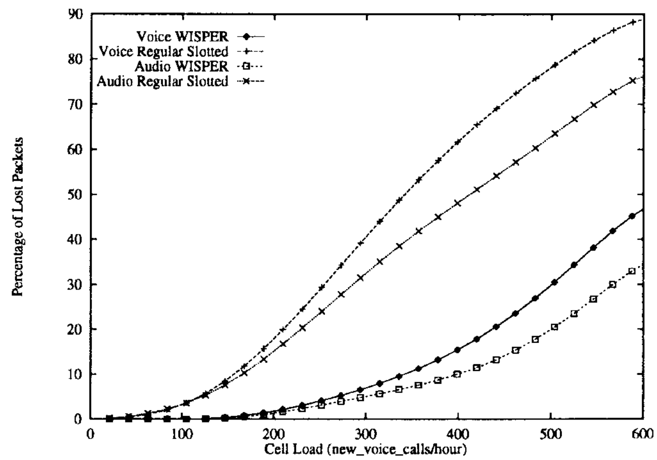
In Fig. 6, we depict the performance results for the WISPER protocol. In Fig. 6(a), we present throughput results for voice, audio, CBR, and VBR video (computer data and e-mail throughputs are not shown because of their small values). In Fig. 6(a), the topmost curve represents the combined throughput of all mobile terminals for all traffic classes in the simulated cell. By far, most of the mobile terminals in the cell transmit voice traffic. However, because of their relatively low transmission rate, their total combined throughput is the smallest. In this experiment, the most bandwidth demanding traffic classes are VBR and CBR video, followed by audio. In Fig. 6(b), we present the average effective packet delay for voice, audio, and video users, i.e., the delays are based on only those packets that were successfully delivered. Voice packets, having the smallest timeout specification (equal to two frames), have the smallest average delay value. In Fig. 6(c), we present the average packet losses for the traffic classes under study. Voice packets have the worst average packet loss

TABLE IV
RELATIVE ARRIVAL RATES OF MOBILES TRANSMITTING THE DIFFERENT TRAFFIC CLASSES

Traffic Class	Percentage
Voice Traffic	70 %
CBR Audio Traffic	3 %
CBR Video Traffic	4 %
VBR Video Traffic	4 %
Data Traffic	15 %
E-mail Traffic	4 %



(a)



(b)

Fig. 7. Performance comparison for WISPER versus regular CDMA. (a) Composite cell throughput. (b) Average packet loss.

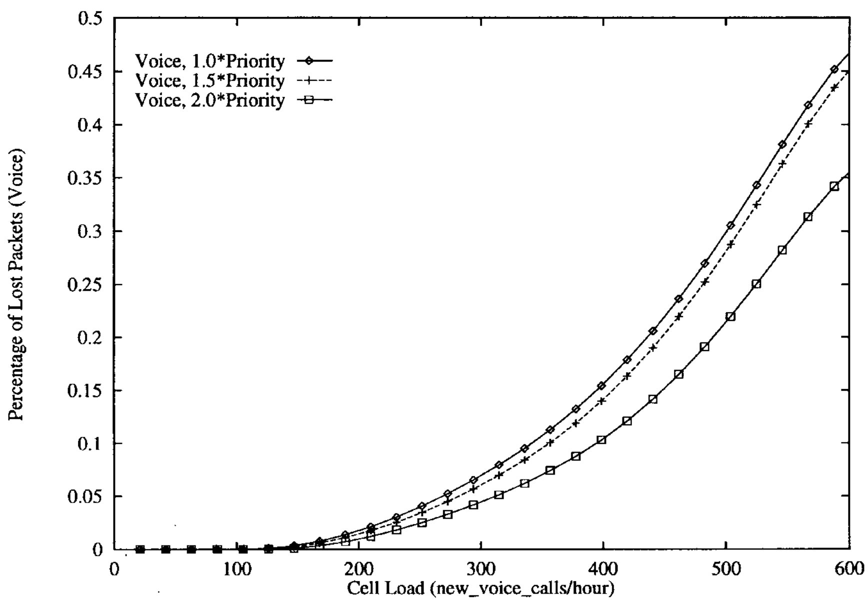


Fig. 8. Average packet loss for voice traffic with different priorities.

characteristic. The reason for this behavior is based on the fact that the maximum transmission rate of voice terminals is limited to only 1 packet/slot, and the maximum priority of a voice packet in any slot cannot be greater than unity, while for the other traffic classes, it can be greater than one. Thus, when congestion occurs, voice packets are the first to be sacrificed. Most applications cannot tolerate packet losses [8], [9] in excess of 1%. For the traffic mix being considered in these simulations, this condition occurs when the arrival rate of new voice calls is about 220 calls/h. Finally, in Fig. 6(d) we present the average message delivery time for computer data and e-mail messages. For acceptable cell-loading conditions, i.e., new voice call-arrival rates less than or equal to about 220 calls/h, the delivery time for the average data messages is excellent, in the order of 1 s.

Fig. 7 shows the performance improvements achieved by our WISPER protocol compared to a regular slotted-CDMA protocol using the conventional power control scheme extended to multimedia traffic [23], where different power levels are set up for different traffic types based on their BER target and the received power is maintained at a fixed level for each traffic type at the base station. The simulation is performed in the link environment of broadband CDMA described in [11]. For simulation parameters, the power control period is 1.25 ms, and the nominal power control step is 1 dB. The pilot signal is transmitted in the control slot of the downlink frame by the base station. The wireless channel is modeled as a Rayleigh fading channel with white Gaussian noise. The regular slotted-CDMA protocol employs the basic FCFS service discipline without BER scheduling to determine the transmission order of packets.

In Fig. 7(a), we present the throughput results for these protocols, considering composite traffic. As the cell load increases, the WISPER protocol provides higher throughput than the regular slotted CDMA. For example, when the cell load is 600 calls/h, it improves the throughput by 30% with respect to the regular slotted CDMA, which is significant. In Fig. 7(b), we show the average packet loss curves for the two protocols. As mentioned before, most applications require a packet loss rate that does not exceed 1% of packet losses. For the regular slotted CDMA protocol, voice packet losses exceed the 1% threshold at cell loads higher than 100 voice calls/h. In contrast, the WISPER protocol can support cell loads higher than 200 voice calls/h. In summary, since the regular slotted CDMA uses different power levels for different traffic types in each slot, they will mutually interfere with each other, thereby reducing the throughput as well as increasing the packet loss.

In the WISPER protocol, it is relatively simple to improve the performance for a certain traffic class or individual mobile terminals. We can improve the performance of a mobile terminal or group of mobile terminals by modifying the priorities of the packets that they generate. Obviously, improving the performance of some users imply that other users will suffer a degradation in their performance measures. In Fig. 8, we present the improvements in the average packet loss characteristics of voice traffic, when the priority of these packets is increased by a factor of 1.5 and 2. Changing the priorities of the packets produced by certain mobiles can be used to fine-tune the performance of the WISPER protocol.

VI. Conclusion

In this paper, we proposed a new MAC protocol called WISPER for CDMA-based wireless multimedia networks. WISPER is a highly flexible protocol that can support different traffic classes that have a wide variety of characteristics and requirements. WISPER is a reservation-based protocol. Since the bandwidth is the most precious resource in a wireless communication system, our emphasis is to design a protocol that maximizes the throughput as well as minimizes the packet losses at the same time.

The presented protocol incorporates a novel packet scheduler which performs the selection and efficient accommodation of the packets to be transmitted in the uplink frames. For each new frame, the packet scheduler prioritizes packet transmissions and accommodates the higher priority packets in the frame, so that packets with equal or similar BER requirements are transmitted in the same slots, thereby maximizing the throughput. Transmission order is determined according to the packets' timeout values and the number of packets ready for transmission at each mobile terminal.

We have evaluated the performance of the protocol under a variety of conditions. We have also compared our protocol to a regular slotted-CDMA protocol using the conventional power control scheme with a different power level for each traffic type. The performance results show that the WISPER protocol provides significant improvement in throughput and packet loss with respect to the regular slotted-CDMA protocol. Furthermore, our protocol is simple to implement in that only one power level can be used for each slot rather than several power levels depending on the number of traffic classes, because the same BER packets are transmitted in the same slot.

References

[1] M. S. Alencar and I. F. Blake, "The capacity for a discrete-state code division multiple-access channel," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 925–937, June 1994.

[2] H. Azad and A. Aghvami, "Multirate spread spectrum direct sequence CDMA techniques," in *IEE Colloq. Spread Spectrum Techniques for Radio Communication Systems, Dig. 1994/098*, Apr. 1994, pp. 4/1–4/5.

[3] A. E. Brand and H. Aghvami, "Multidimensional PRMA with prioritized Bayesian broadcast: A MAC strategy for multiservice traffic over UMTS," *IEEE Trans. Veh. Technol.*, vol. 47, pp. 1148–1161, Nov. 1998.

[4] M. Decina and T. Toniatti, "Bandwidth allocation and selective discarding for a variable bit rate video and bursty data calls in ATM networks,," *Int. J. Digital Analog Commun. Syst.*, vol. 5, no. 2, pp. 85–96, Apr.–June 1992.

[5] N. Doi, T. Yano, and N. Kobayashi, "DS/CDMA prototype system transmitting low bit-rate voice and high bit-rate ISDN signals," in *Proc. IEEE Vehicular Technology Conf.*, vol. 1, 1994, pp. 51–54.

[6] K. S. Gilhousen, I. M. Jacobs, R. Padovani, A. J. Viterbi, L. A. Weaver, and C. E. Wheatley III, "On the capacity of a cellular CDMA system," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 303–312, May 1991.

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

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

[9] J. Gruber and L. Strawczynski, "Subjective effects of variable delay and speech clipping in dynamically managed voice systems," *IEEE Trans. Commun.*, vol. COM-33, pp. 801–808, 1985.

[10] T. Hu and M. K. Liu, "Power control for wireless multimedia CDMA systems," *Electron. Lett.*, vol. 33, no. 8, pp. 660–662, Apr. 1997.

[11] R. Kohno, R. Meidan, and L. B. Milstein, "Spread spectrum access methods for wireless communications," *IEEE Commun. Mag.*, pp. 58–67, Jan. 1995.

[12] C. Lin I, R. D. Gitlin, "Multi-code CDMA wireless personal communications networks," in *Proc. IEEE Int. Conf. Commun. ICC'95*, June 1995, pp. 1060–1064.

[13] Z. Liu, M. E. Zarki, M. J. Karol, and K. Y. Eng, "A demand-assignment access control for multi-code DS-CDMA wireless packet (ATM) networks," in *Proc. IEEE Int. Conf. Communications ICC'96*, June 1996.

[14] T. Ojanperä and R. Prasad, "An overview of air interface multiple access for IMT 2000/UMTS," *IEEE Commun. Mag.*, vol. 36, pp. 82–95, Sept. 1998.

[15] N. Passas, S. Paskalis, D. Vali, and L. Merakos, "Quality of service oriented medium access control for wireless ATM networks," *IEEE Commun. Mag.*, vol. 35, pp. 42–52, Nov. 1997.

[16] R. Pichna and Q. Wang, "A medium access control protocol for a cellular packet CDMA carrying multirate traffic," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 1728–1736, Dec. 1996.

[17] M. B. Pursley, D. V. Sarwate, and W. E. Stark, "Error probability for direct-sequence spread-spectrum multiple-access communications–Part I: Upper and lower bounds," *IEEE Trans. Commun.*, vol. COM-30, May 1982.

[18] D. Raychaudhuri, "ATM-based transport architecture for multiservices wireless personal communications," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 1401–1414, Oct. 1994.

[19] C. Roobol, P. Beming, J. Lundsjo, and M. Johanson, "A proposal for a MAC protocol for wideband CDMA capable of handling real-time and nonreal time services," in *Proc. IEEE Veh. Technol. Conf.*, 1998.

[20] R. R. Roy, "Networking constraints in multimedia conferencing and the role of ATM networks," *AT&T Tech. J.*, vol. 73, no. 4, pp. 97–108, July–Aug. 1994.

[21] P. F. M. Smulders and C. Blondia, "A MAC protocol for ATM-based indoor radio networks," *European Co-Operation in the Field of Scientific and Technical Research (EURO-COST)*, COST 231 TD (94), Apr. 1994.

[22] G. L. Stüber, "Performance analysis of slotted direct-sequence spread-spectrum multiple-access networks," in *Proc. IEEE MILCOM'88*, pp. 833–837.

[23] J. T. Wu and E. Geraniotis, "Power control in multimedia CDMA networks," in *Proc. IEEE Veh. Technol. Conf.*, 1995, pp. 789–793.

[24] J. Wu and R. Kohno, "A wireless multimedia CDMA system based on transmission power control," *IEEE J. Select. Areas Commun.*, vol 14, pp. 683–691, May 1996.

[25] R. Wyrwas, W. Zhang, M. J. Miller, and R. Anjaria, "Multiple access options for multi-media wireless systems," *Wireless Commun., Future Directions*, pp. 305–317, Apr. 1992.

[26] W.-B. Yang and E. Geraniotis, “Admission policies for integrated voice and data traffic in CDMA packet radio networks,” *IEEE J. Select. Areas Commun.*, vol 12, pp. 654–664, May 1994.

[27] S. Yao and E. Geraniotis, “Optimal power control law for multimedia multirate CDMA systems,” in *Proc. IEEE Veh. Technol. Conf.*, 1996, pp. 392–396.

[28] L. C. Yun and D. G. Messerschmitt, “Variable quality of service in CDMA systems by statistical power control,” in *Proc. IEEE Int. Conf. Communications ICC’95*, June 1995, pp. 713–719.

[29] Z. Zhang and Y.-J. Liu, “Performance analysis of multiple access protocols for CDMA cellular and personal communication services,” in *Proc. IEEE INFOCOM’93*, vol. 3, pp. 1214–1221, 1993.

[30] H. Zhang, “Service disciplines for guaranteed performance service packet-switching networks,” *Proc. IEEE*, vol. 83, pp. 1374–1399, Oct. 1995.

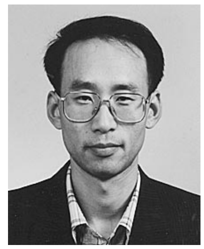
Ian F. Akyildiz (F’95), received the B.S., M.S., and Ph.D. degrees in computer engineering from the University of Erlangen-Nuremberg, Germany, in 1978, 1981, and 1984, respectively.

He is currently a Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, where he also serves as the Chair of the Telecommunications Area, and is Director of Broadband and Wireless Networking Laboratory. He has held Visiting Professorships at the Universidad Tecnica Federico Santa Maria, Chile, Universite Pierre et Marie Curie (Paris VI), Ecole Nationale Superieure Telecommunications, Paris, France, Universidad Politecnico de Cataluna in Barcelona, Spain, and Universidad Illes Baleares, Palma de Mallorca, Spain. He has published over 200 technical papers in journals and conference proceedings. His current research interests are in wireless networks, satellite networks, ATM networks, and multimedia communication systems.

Dr. Akyildiz is Fellow of the Association of Computing Machinery (ACM). He served as a National Lecturer for ACM from 1989 until 1998. He received the ACM Outstanding Distinguished Lecturer Award in 1994. He received the Don Federico Santa Maria Medal for his services to the Universidad of Federico Santa Maria in Chile. He also received the 1997 IEEE Leonard G. Abraham Prize award for his paper entitled “Multimedia Group Synchronization Protocols for Integrated Services Architectures” published in the IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS in January 1996. He is an Editor for IEEE/ACM TRANSACTIONS ON NETWORKING, *Computer Networks Journal*, *ACM/Springer Journal for Multimedia Systems*, *ACM/Baltzer Journal of Wireless Networks*, and *Journal of Cluster Computing*. He is a past Editor for IEEE TRANSACTIONS ON COMPUTERS (1992–1996). He was the Program Chair of the 9th IEEE Computer Communications Workshop in 1994. He also served as the Program Chair for the ACM/IEEE Mobile Computing and Networking (MOBICOM’96) Conference and the IEEE INFOCOM’98 Conference.

David A. Levine received the B.S. degree in electronics and communications from the Universidad La Salle, Mexico City, in 1988, the M.S.E.E. degree from the University of Virginia, Charlottesville, in 1991, and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, in 1996.

While at Georgia Tech, he was the recipient of a scholarship from the National Science Foundation. In the summer of 1995, he worked at AT&T Bell Laboratories, Whippany, NJ, as a summer intern. He is currently a Member of Technical Staff at Bell South Telecommunications, Atlanta, GA. His current research interests are in wireless networks, optical local area networks, performance evaluation, and computer telephony.



Inwhee Joe received the B.S. and M.S. degrees in electronics engineering from Hanyang University, Seoul, Korea, the M.S. degree in electrical and computer engineering from the University of Arizona, Tucson, in 1994, and the Ph.D. degree in electrical and computer engineering from Georgia Institute of Technology, Atlanta, GA, in 1998.

From 1985 to 1992, he was an Engineer in the Research Center of DACOM Corporation, where his research focused on networking software, operating systems, and distributed systems. Since 1998, he has been a Member of the Network Research Group at Oak Ridge National Laboratory, Oak Ridge, TN. His current research interests include wireless ATM networks, mobile networking, multimedia networking, and performance evaluation.