Wireless Networks 7, 43–54, 2001

Highly Efficient Voice–Data Integration over Medium and High Capacity Wireless TDMA Channels

P. KOUTSAKIS and M. PATERAKIS
Department of Electronics and Computer Engineering, Technical University of Crete, 73100 Chania, Greece

Abstract. A new medium access control (MAC) protocol for mobile wireless communications is presented and investigated. We explore, via an extensive simulation study, the performance of the protocol when integrating voice and data traffic over two wireless channels, one of medium capacity (referring mostly to outdoor microcellular environments) and one of high capacity (referring to an indoor microcellular environment). Data message arrivals are assumed to occur according to a Poisson process and to vary in length according to a geometric distribution. We evaluate the voice packet dropping probability and access delay, as well as the data packet access and data message transmission delays for various voice and data load conditions. By combining two novel ideas of ours with two useful ideas which have been proposed in other MAC schemes, we are able to remarkably improve the efficiency of a previously proposed MAC scheme [5], and obtain very high voice sources multiplexing results along with most satisfactory voice and data performance and quality of service (QoS) requirements servicing. Our two novel ideas are the sharing of certain request slots among voice and data terminals with priority given to voice, and the use of a fully dynamic low-voice-load mechanism.

Keywords: MAC protocol, voice–data integration, QoS

1. Introduction

Future generation wireless personal communication networks (PCN) are expected to provide multimedia capable wireless extensions of fixed ATM/B-ISDN, as data and video traffic will soon gain in importance due to the continuous proliferation of small, portable and inexpensive computing devices. This is the ultimate goal of wireless communication, to allow the user access to the capabilities of the global network at any time without regard to location or mobility.

Current and future wireless networks are and will be based on the cellular concept. System capacity can be increased by:

(a) using a cellular structure with a cell size as small as possible (microcells) to increase frequency reuse. Microcell diameters are usually of the order of a few hundred meters, therefore the round-trip propagation delay within a microcell is negligible (of the order of 1 μs);

(b) using efficient medium access control (MAC) protocols to exploit the variations in access and service required by disparate sources.

A well-designed multiple access protocol will reduce system costs by maximizing system capacity, integrating different classes of traffic (e.g., voice, data and video, as opposed to today's picture, where wireless networks are optimized for voice communications only), and satisfying the diverse and usually contradictory quality of service (QoS) requirements of each traffic class (such as voice packet dropping probability, voice packet access delay, data packet access delay, data message delay, video packet dropping probability) whilst apportioning the limited radio channel bandwidth among them.

In this work, we design and evaluate multiple access schemes that multiplex voice traffic at the vocal activity (talkspurt) level to efficiently integrate voice and data traffic in outdoor (medium capacity channel) and indoor (high capacity channel) microcellular environments.

Within the microcell, spatially dispersed source terminals share a radio channel that connects them to a fixed base station. The base station allocates channel resources, delivers feedback information and serves as an interface to the mobile switching center (MSC). The MSC provides access to the fixed network infrastructure. Since the base station is the sole transmitter on the downlink channel, it is in complete control of the downstream traffic, using Time Division Multiple Access (TDMA) to relay information to the users. Thus, we focus on the uplink (mobiles to base station) channel, where a MAC scheme is required in order to resolve the source terminals contention for channel access.

We assume that voice and data traffic is generated by mobile users who access the network with small, lightweight and low-power devices (i.e., in the category of low tier PCS [7]). Speech alternates between periods of talk (talkspurts) and silence. Thus, voice terminals only require channel access during talkspurt and the time periods corresponding to silence gaps within a conversation can be used to transmit packets from other source terminals (i.e., multiplexing occurs at the talkspurt level). Voice packet delay requirements are stricter than those for data packets, because delays in voice communication are annoying to a listener. Thus, each voice packet must be delivered within a specified maximum delay. Whenever the delay experienced by a voice packet exceeds this maximum delay, the voice packet is dropped. Speech can withstand a small (1–2%) amount of dropped packets without suffering large quality degradation [14], at least one which can be perceived by humans. On the other hand, data applications are more tolerant of delays (delays of up to 200 ms are often acceptable), but 100% delivery of correct packets is often required (e.g., in the case of a file transfer) [3].

The paper is organized as follows. In section 2 we present the two channel models, the voice and data traffic models, and we analyze the transmission protocols for voice and data traffic. In section 3 we introduce the experimental system design parameters. Section 4 includes our simulation results along with discussion upon them and comparisons with other, previously proposed MAC schemes. Finally, section 5 contains our concluding remarks.

2. System model

In this section, we present the structure of the channel frame, the actions of the voice and data terminals, and the scheduling algorithm followed by the base station (BS). Additionally, we explain the data and voice transmission protocols, as well as their difference, and we introduce the voice and data traffic models.

2.1. Channel frame structure

The uplink channel time is divided into time frames of equal length. The frame duration is selected such that a voice terminal in talkspurt generates exactly one packet per frame. As shown in figure 1 (which presents an example of the channel frame structure), each frame consists of three *types* of intervals. These are the *voice request* intervals, the *data request* intervals and the *information* intervals.

Within an information interval, each slot accommodates exactly one, fixed length, packet that contains voice or data information and a header. All request intervals (voice or data) are subdivided into mini-slots and each mini-slot accommodates exactly one, fixed length, request packet. For both voice and data traffic, the request must include a source identifier. For data traffic, the request must also include message length in packets and perhaps quality of service parameters such as priority and required slots/frame. We assume that both types of request intervals contain an equal number of mini-slots, and we distribute the data request intervals uniformly within the frame. This way, since data message arrivals occur uniformly within the frame duration (they are assumed to be Poisson), we allow the data terminals to transmit their requests soon after their messages have been generated.

Since we assume that all of the voice transitions occur at the frame boundaries, we place all voice request intervals at the beginning of the frame, in order to minimize the voice packet access delay.

The voice and data terminals do not exhaust their attempts for a reservation within the request intervals. Any other free, at the time, information slot of the frame can be temporarily used as an extra request slot (ER slots) for both voice and data terminals, with priority given to the voice terminals. Each one of the ER slots is further divided into minislots, each being able to accommodate exactly one request packet, just like a standard request (R) slot. This approach is introduced and implemented in [12,13].

By using more than one minislot per request slot, a more efficient usage of the available request bandwidth is possible. *We introduce the idea that certain request slots can be shared by voice and data terminals* (*first by voice terminals and, after the end of voice contention, by data terminals*), in order to optimize the use of the request bandwidth. The concept of reserving a minimum bandwidth for both voice and data terminals to make reservations helps to keep the access delay within relatively low limits and gives clearly better performance than the PRMA [15], where the absence of request slots leads to a continuously decreasing probability of finding available information slots as traffic increases, and hence, to greater access delays. A request bandwidth of 2–3% is usually sufficient for high system performance (in the sense that it suffices for the requesting terminals to transmit their requests, while at the same time it does not consume a large portion of the bandwidth and leaves enough "space" for terminals with a reservation to transmit their information packets).

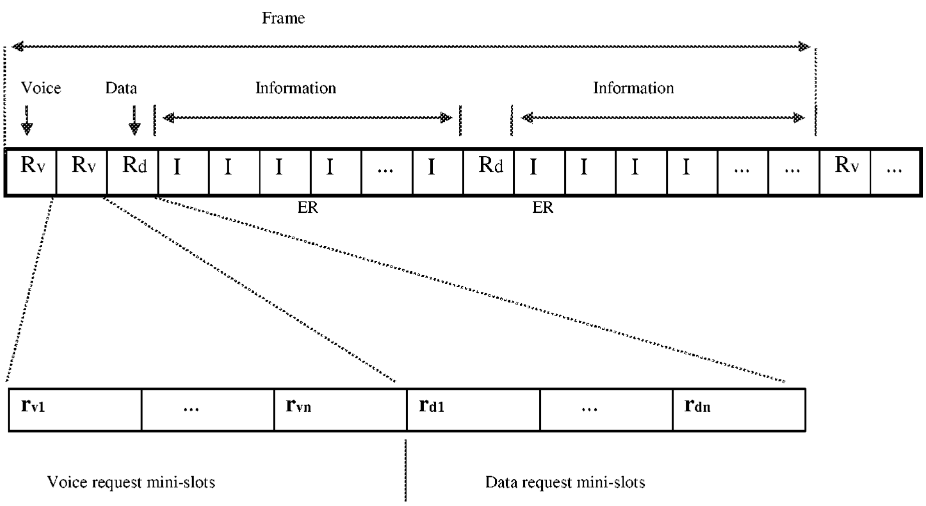


Figure 1. An example of a channel frame structure showing the voice, data and information intervals within a frame.

2.2. Actions of voice and data terminals, and base station scheduling

Voice and data terminals with packets, and no reservation, contend for channel resources using a random access protocol to transmit their request packets only during the voice or data, respectively, request intervals. The base station broadcasts a short binary feedback packet at the end of each mini-slot indicating only the presence or absence of a collision within the mini-slot (collision (C) versus non-collision (NC)). Since the feedback packet is short (several bits) and the propagation delay within a microcell is negligible, we assume that the feedback information is immediately available to the terminals (i.e., before the next mini-slot). Upon successfully transmitting a request packet the terminal waits *until the end of the corresponding request interval* to learn of its reservation slot (or slots). If unsuccessful within the request intervals of the current frame, the terminal attempts again in the request intervals of the next frame. A terminal with a reservation transmits freely within its reserved slot. Generally, a terminal that fails to transmit a request tries again in successive frames until it succeeds. However, since voice packets that age beyond the voice delay limit are dropped, a voice terminal may stop transmitting requests without ever succeeding, because all of its packets have timed out and it has transitioned into silence.

To allocate channel resources, the BS maintains a dynamic table of the active terminals within the microcell. For example, information within the table might include the terminal identifier, the virtual circuit identifier, the channel resources allocated, and quality of service parameters. Upon successful receipt of a request packet, the BS provides an acknowledgment and queues the request. The BS allocates channel resources *at the end of the corresponding request interval*, if available. If the resources needed to satisfy a request are unavailable, the request remains queued. Voice and data terminals with queued requests must continuously monitor the base-to-mobile channel. Upon call or message transmission completion, or when an active terminal exits the microcell (handover) the BS will delete the table entry after some prescribed period of time (the state transitions are shown in figure 2).

As we focus on steady state channel access, we do not address call set-up and tear down issues. We assume that the BS always allocates the earliest available information slot within the frame, and that voice is of higher priority than data traffic. Thus, the BS services every outstanding voice request before servicing any data requests. Within each priority class, the queuing discipline is assumed to be First Come First Served (FCFS).

Finally, *we apply a low-voice-load mechanism to our scheme*. As data terminals try to transmit messages that vary in length and are, on average, much longer than one packet (424 bits), it would be both unfair to them and diminishing to our system's performance to not allocate to them more than one slot per frame (which is exactly enough for voice terminals) if resources (slots) were available. On the other hand,

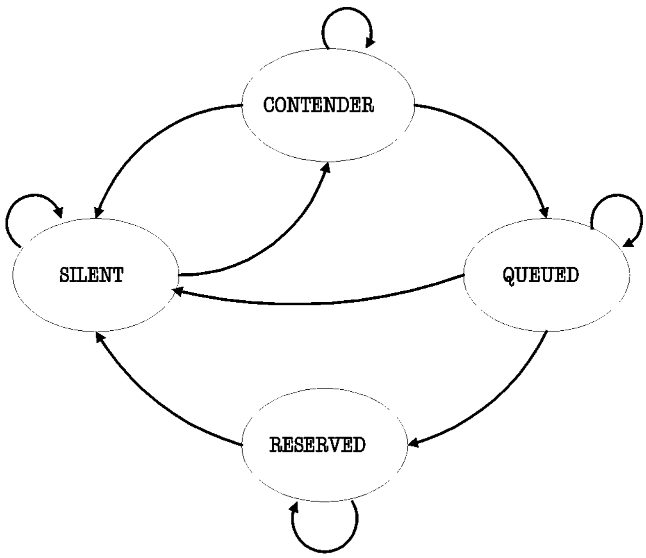


Figure 2. State transition diagram for an active voice terminal.

by allocating more than one slot per frame to data terminals, voice terminals would find a lower number of information slots available for either reservations or requests (ER slots), and our objective is, as already pointed out, to enforce voice priority. Therefore, we introduce the following mechanism.

We define the *frame voice occupancy* as the ratio of

$$\frac{\text{voice reservations} + \text{voice requests}}{\text{total number of information slots in the frame}}.$$

This ratio is calculated by the BS immediately after the end of the voice request slots of each frame. If the ratio is lower than a set limit, which means that voice activity will be low in the current frame, we allow data terminals with requests to acquire more than one slot *in the current frame*. Notice however, that this allocation policy is temporary and only one (the first allocated) slot is guaranteed to the data terminals in subsequent frames. The BS will immediately deallocate all but the first slots of the data terminals if the frame voice occupancy ratio exceeds the set limit. If, on the other hand, a low voice load situation continues to exist, the data terminal may keep in the next frame as many of the slots it reserved as it needs in order to transmit the remaining packets of its message.

The selection of the *low frame voice occupancy limit* and of the *maximum number of slots that can be allocated to data terminals within a frame* (*these are the two parameters of the low-load mechanism*) must be done carefully, so that even in the case of low voice load enough information slots will still remain available in the current frame for voice terminals who enter talkspurt to use as ER slots.

More specifically, these selections should be based on the combination of the following two factors:

(a) The average data message length (according to the data traffic model); if possible, the maximum possible number of allocated slots should approach or be equal to this length.

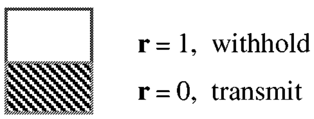


Figure 3. Visualizing the two-cell stack algorithm.

(b) The “if possible” notion in the previous factor is related to the existence or not of the capability to accommodate within the frame a few average-sized data messages. Therefore, the second factor is the channel capacity. In the case, for example, of a low capacity channel (e.g., with 20 information slots per channel frame) and data messages of an average length equal to 10 packets, it is clear that allocating 10 slots to a requesting data terminal would most probably result both in an absence of ER slots for possible use by voice terminals and in enormous data packet access delays, as the data terminals whose requests lie at the bottom of the BS’s data request queue will have to wait unacceptably long to transmit their first packet.

(c) The expected data load.

In the cases of the two wireless channels in our study, we will introduce and explain our numerical choices for the two parameters of the low-load mechanism in section 3.

2.3. Transmission protocols

2.3.1. Data terminals

The *two-cell stack* blocked access collision resolution algorithm [16,17] is adopted for use by the data terminals in order to transmit data request packets. This algorithm is of window type, with FCFS-like service.

The operation of the collision resolution mechanism of this protocol can be visualized by a two-cell stack (as shown in figure 3), where in a given data request minislot the bottom cell contains the transmitting terminals, and the top cell contains the withholding terminals. If the transmitting set contains more than one terminal, it is split probabilistically into two subsets (with equal probability), one of which remains in the transmitting cell while the other one joins the withholding cell. The end of the terminal contention for the request minislots is uniquely identified by the occurrence of two consecutive non-collisions. For more details on window type collision resolution algorithms the interested reader is referred to [1,18].

2.3.2. Voice terminals

Quite a few reservation random access algorithms have been proposed in the literature, for use by contending voice terminals to access the channel (e.g., PRMA [15], two-cell stack [3,4], controlled ALOHA [1,3], three-cell stack [5]). In our study, we adopt the *two-cell stack* reservation random access algorithm, due to its operational simplicity, stability and relatively high throughput when compared to the PRMA (ALOHA) [6] algorithm. This protocol differs from the two-cell stack data collision resolution protocol in 2.3.1, in that the contending set of voice terminals is split probabilistically into two equiprobable subsets *at the beginning of a frame*.[1] Only one of these subsets is transmitted in the first voice request minislot. The end of the voice contention is again uniquely identified by the occurence of two consecutive non-collisions.

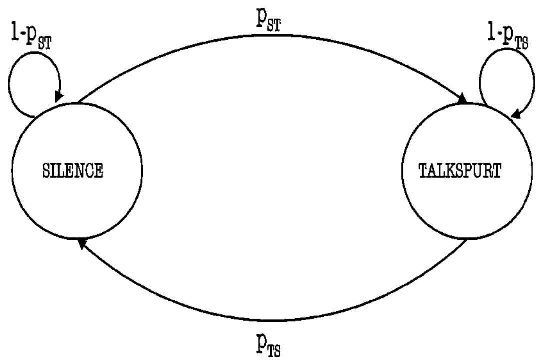


Figure 4. The voice source activity discrete time Markov chain model.

2.4. Voice traffic model

Our primary voice traffic model assumptions are the following:

(1) Voice terminals are equipped with a voice activity detector [2,15]. Voice sources follow an alternating pattern of talkspurts and silence periods (on and off), and the output of the voice activity detector is modeled by a two-state discrete time Markov chain as shown in figure 4.

(2) The number of active voice terminals, N, in the system is assumed to be constant over the period of interest. This is because the changes in the number of calls are usually on the order of tens of seconds, while the frame duration is on the order of tens of milliseconds [19].

(3) All of the voice transitions (e.g., talk to silence) occur at the frame boundaries. This assumption is reasonably accurate, taking into consideration that the duration of a frame is equal to 12 ms, while the average duration of the talkspurt and silence periods exceeds 1 s.

(4) The voice delay limit is equal to the duration of two channel frames (i.e., 24 ms).

(5) The channel is error-free and without capture.

(6) Reserved slots are deallocated immediately. This implies that a voice terminal holding a reservation signals the BS upon the completion of its talkspurt.

2.5. Data traffic model

We adopt the following data traffic model.

Data traffic has low priority compared to voice traffic, and data messages are generated by a large unknown number of

[1] Notice that the voice terminals, unlike data terminals, split before the beginning of the voice contention. Our results for the algorithm without this initial split showed the voice capacity to be lower by several terminals.

data terminals (theoretically infinite). The aggregate message arrivals are Poisson distributed with mean λ messages per frame, while the messages vary in length according to a geometric distribution with parameter q and mean $B = 1/q$. B is expressed in packets per message, and the steady state data rate packet arrival is equal to λB packets/frame.

3. System parameters

We use computer simulations to study the performance of our two MAC schemes. The simulations were conducted with the parameters contained in table 1. Each simulation point is the result of an average of 10 independent runs, each simulating 305000 frames (the first 5000 of which are used as warmup period).

The differences between certain parameters of the two schemes concern the channel rate, the total number of slots per frame and the number of request slots per frame. More specifically:

(a) The channel rate for the medium capacity channel is 1.8 Mbps (from [7–9]), whereas for the high capacity channel is 9.045 Mbps (from [10]).

(b) In the medium capacity channel, the 12 ms of frame duration accommodate approximately 51 slots. To account for guard time and synchronization, we assume that 50 slots are available per frame. As shown in figure 5, in this scheme the first slot is used as a request slot. The request slot is subdivided into six minislots. *Voice terminals get absolute access priority within the minislots* (*i.e., data terminals can transmit their request packets only after the voice contention has ended*).
As it is easily understood, for the medium capacity channel, the rather limited bandwidth and consequently the small number of slots/frame, does not allow us to fully implement our ideas introduced in section 2. More specifically, it is not possible to implement the idea of uniform data request slot distribution within the frame. Fortunately, we are able to apply all our ideas in the case of the high capacity channel.

(c) In the high capacity channel, the 12 ms of frame duration accommodate 256 slots. As shown in figure 6, in this scheme six slots are used as request slots. This corresponds to a 2.34% request bandwidth, which is within the desired range (2–3%). The first four slots of the frame compose the first request interval. *Voice terminals again get absolute access priority within the minislots of this request interval.* The other two request slots, which are exclusively dedicated to data request intervals, are slots 88 and 172 of the frame. Thus, the data request slots (i.e., these two plus the potential use of the minislots that the voice contention left unused in the beginning of the frame) are uniformly distributed throughout the frame.

The common parameters of the two schemes are shown in table 1. We should note that:

(1) In our design, we chose the number of minislots per request interval (6 for the medium capacity channel and 4 for the high capacity channel), to allow for guard time and synchronization overheads, for the transmission of a generic request packet (e.g., 40 bits long) that contains

Table 1
Experimental system parameters.

Design parameters	Medium capacity channel	High capacity channel
Channel rate (Mbps)	1.8	9.045
Speech codec rate (Kbps)	32	32
Frame duration (ms)	12	12
Slots per frame	50	256
Slot duration (μs)	240	46.875
Request slots per frame	1	6
Minislots per request slot	6	4
Packet size (bytes)	53 (5 header)	53 (5 header)
Voice delay limit (ms)	24	24
Mean talkspurt duration (s)	1.41	1.41
Mean silence duration (s)	1.78	1.78
Maximum voice dropping probability	0.01	0.01
Low frame voice occupancy limit (slots)	46	238
Average delay limit for data message transmission (ms)	200	200

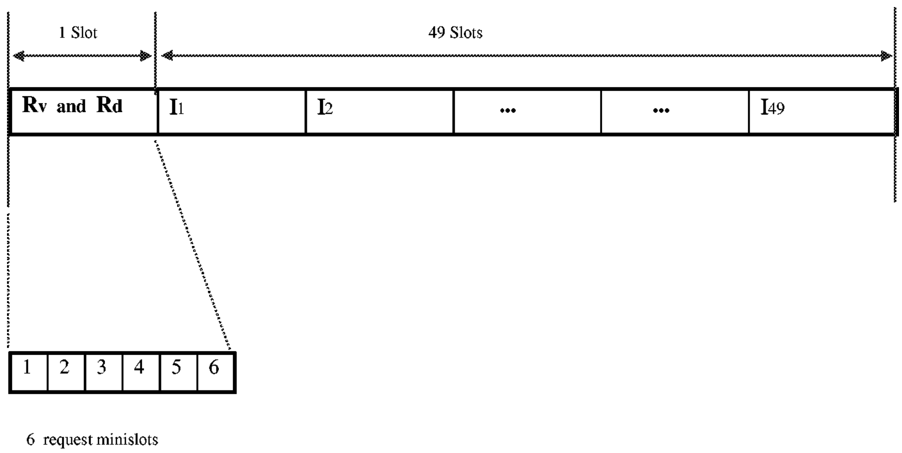


Figure 5. Frame structure for the 1.8 Mbps channel.

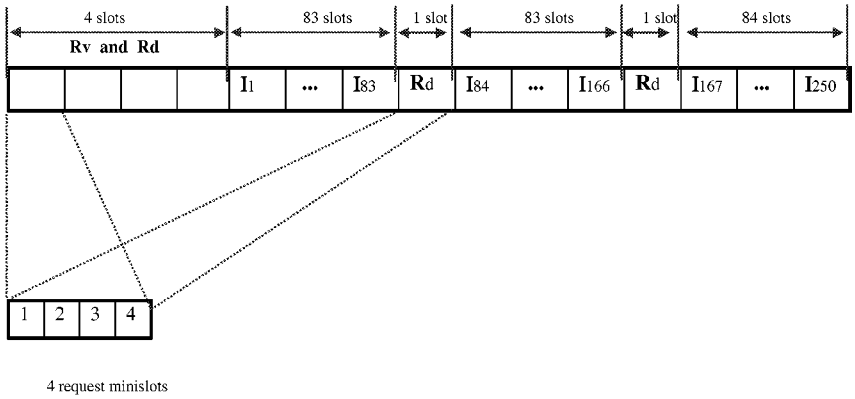


Figure 6. Frame structure for the 9.045 Mbps channel.

the source identifier, along with some data (e.g., priority, slots required, etc.), and for the propagation delay within the microcell.

(2) Because of assumption 3 of our voice traffic model, all voice request intervals are located at the beginning of each frame.

(3) Parameter q of the data traffic model is equal to $1/8$, thus, $B = 8$. Therefore, the average data message is assumed rather short since it contains approximately 3400 bits.

(4) According to the low-load mechanism described in section 2, we set the low frame voice occupancy limit at 95% (this value was chosen via simulation). In case the frame voice occupancy is below the above limit, the BS allocates to the data terminals which have successfully transmitted their request packets up to 8 slots (if the data terminal has requested that many in its request packet). The choice of the two parameters of the low-load mechanism (95%, 8) is reasonable, as it permits, even for a high λ (e.g., 9 data messages/frame in the high capacity channel) the existence of ER slots for possible use by voice terminals, preserving this way the voice priority we want to enforce. This parameters choice is further supported by the simulation results we present in section 4.

4. Results and discussion

This section is divided in two parts. The first part refers to the scheme for the medium capacity channel, which we name VDI-MCC (Voice–Data Integration in a Medium Capacity Channel). The second part refers to the scheme for the high capacity channel, which we name VDI-HCC (Voice–Data Integration in a High Capacity Channel).

4.1. VDI-MCC

To demonstrate the very good performance of our scheme, we will compare it with two previously proposed efficient schemes for voice–data integration, IPRMA [20] and RRA [3,6]. The comparison with IPRMA can only be done conceptually, since the system parameters of the two schemes are completely different, whereas in the comparison with RRA we use the same system parameters.

In IPRMA, speech users are allowed to contend for reservation slots on a frame-by-frame basis, while data users may reserve multiple slots across a frame to increase throughput. The protocol includes a priority mechanism which ensures that speech users have greater access to idle slots.

The IPRMA protocol presents four disadvantages when compared to VDI-MCC. The first disadvantage, as already stated in section 2.3.1, is the use of the PRMA algorithm to resolve voice terminal contention, as opposed to our use of the two-cell stack algorithm. PRMA is an ALOHA-based reservation random access algorithm with constant retransmission probability. As such, it exhibits instability for high loads and achieves lower throughput than the inherently stable tree and stack collision resolution algorithms [1]. The second disadvantage is the absence of request slots, the importance of which has been stated earlier.[2]

The other two, more important disadvantages of IPRMA are that it does not provide absolute priority to voice traffic over data traffic, and the static nature of its low-load mechanism. The granting, in IPRMA, of a much smaller transmission probability to the data terminals is not enough to ensure absolute voice priority, which, however, is guaranteed in VDI-MCC. As for the IPRMA's low-load mechanism, the following remarks should be made. In IPRMA, if there are k idle slots in a frame of N slots, the authors impose a speech priority of M slots and a data user who has several packets ready for transmission is allowed to reserve up to $(k - M - 1)$ slots, thereby keeping a minimum number of idle slots available for speech transmission. This handling of the low voice load situation presents the innate problem of the external parameter M imposition, instead of its dynamic adjustment, which would be best. Imposing the M limitation externally will certainly result, in some frames, in sacrific-

[2] However, it should be noted that for the simulation parameters considered in [20] (224 Kbps channel transmission rate, 20 slots per frame) the use of even one request slot would incur a 5% bandwidth overhead (1/20 slots) on the system and would potentially deteriorate, instead of improving, the system performance.

Table 2
Steady state voice performance at voice capacity, the 97.5% t-confidence intervals are constructed in the usual way [11].

MAC protocol	N (terminals)	P_{drop} (%)	Mean access delay (ms)	Throughput (packets/frame)	Percentage of lossless talkspurts
VDI-MCC	100	0.930 ± 0.039	21.31 ± 0.54	43.82 ± 0.10	86%
RRA	97	0.917 ± 0.037	30.84 ± 0.54	42.49 ± 0.08	82%

ing slots for use by voice terminals that will not need them. On the contrary, our low-load mechanism is totally dynamic, taking into consideration the voice users needs, the knowledge of which is possible because of the use of the request slots. This way the data users are well served, as it is possible for them to acquire, if needed, all the available information slots in the frame.

The almost obligatory absence of request slots in IPRMA does not justify for the lack of a dynamic low-load mechanism. One easily implemented possible approach would be to make an estimation of the number of voice terminals that will try to transmit in each slot of the frame, based on the number of reserved slots in the frame and the probabilities p_{T} and p_{ST} of the voice source model, in a way similar to that of the Controlled ALOHA algorithm [1,3]. With the use of such an estimation procedure, data users would be able to dynamically acquire the maximum (or quite close to the maximum) number of available slots of each frame.

The RRA protocol, designed and implemented by Cleary et al. [3,6], considered a system model quite similar to ours (same traffic models and system parameters), with five differences:

1. RRA uses *two* request slots, the second of which is used for data requests.
2. RRA does not use ER slots.
3. In RRA, the BS allocates resources to the requesting terminals *at the end of each channel frame*, and not *at the end of the request slot of the current channel frame*, as is done in VDI-MCC.
4. In RRA, in order to achieve absolute voice traffic priority, the BS preempts data reservations to service voice requests. More specifically, whenever new voice requests are received and every information slot within the frame is reserved, the BS attempts to service the voice requests by canceling the appropriate number of reservations belonging to data terminals (if any). When a data reservation is canceled, the BS notifies the affected data terminal and places an appropriate request at the front of the data request queue. With the use of this mechanism, *data traffic does not affect the accomodation and QoS provided to the voice users*.
5. In RRA, *data terminals may acquire only one information slot per frame*.

We compare VDI-MCC with RRA on three types of results, to demonstrate our scheme's significantly better per-

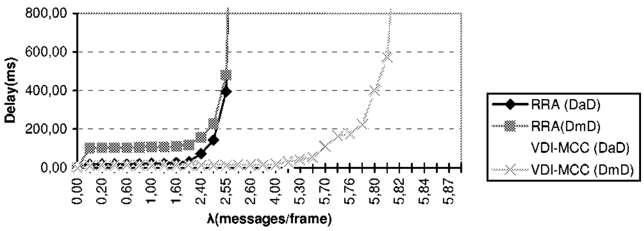


Figure 7. Steady state mean data delays in the absence of voice traffic.

formance. First, we will examine the situation of accommodating voice traffic only, then the situation of accommodating data traffic only and finally the situation of voice–data integration. We will show that VDI-MCC improves the results on the voice capacity and QoS requirements and remarkably improves the results on the data performance and QoS requirements.

4.1.1. Results for voice traffic only

Our results (presented in table 2) show that:

(a) The voice capacity (defined as the *maximum number of VTs for which P_{drop} is less than 1%*) is 100 voice terminals (VTs) for VDI-MCC, corresponding to a multiplexing gain of 2 (100/50), while it is equal to 97 voice terminals for RRA, corresponding to a multiplexing gain of 1.94. Also, our simulation results show that for the same number of VTs, the voice packet dropping probability is always lower in VDI-MCC than in RRA.

(b) VDI-MCC achieves a significantly lower mean voice packet access delay and a higher percentage of lossless talkspurts (i.e., talkspurts with no packet dropping).

4.1.2. Results for data traffic only

As shown in figure 7, and as expected by: (a) the immediate allocation of resources (after the end of the corresponding request interval, while in RRA allocation is done at the end of each channel frame), (b) the use of ER slots, and (c) the possible allocation of more than 1 information slots per frame to the data terminals in low voice load situations, the improvements in mean data packet access delays (DaD)[3] and mean data message transmission delays (DmD)[4] over the corre-

[3] The data packet access delay is defined as the time period from the instant a data terminal generates a message, until it completes the transmission of the first packet of its message in a reserved slot.

[4] The data message delay is defined as the time period from the instant a data terminal generates a message, until it completes the transmission of the last packet of its message in a reserved slot.

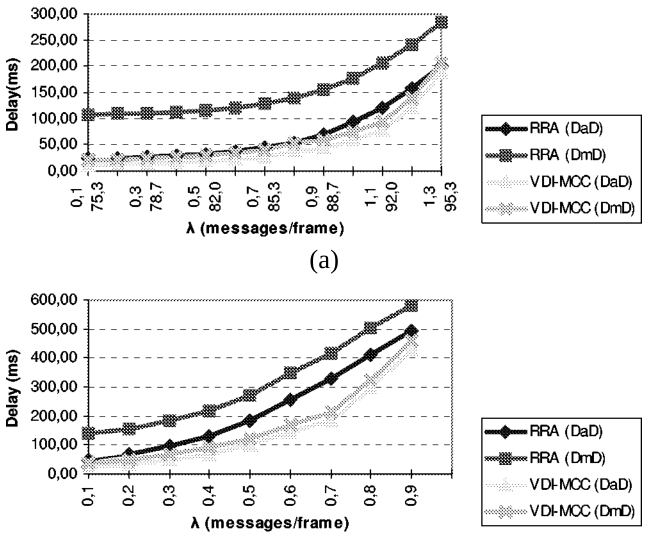
sponding delays for RRA are *dramatic*. Both of these data performance metrics are remarkably lower in VDI-MCC. More specifically, we observe that in RRA the data message delay is consistently greater than the data access delay by about 84 ms. This is because the data message delay in this scheme equals the sum of the access delay and $(B - 1)F$, where F is the frame duration (i.e., 12 ms). On the contrary, in VDI-MCC the data message delay is greater than the data access delay by just 6.5–10 ms, due to the three basic improvements mentioned above. The minimum difference of 6.5 ms between the two delays in our scheme is explained by the fact that, although data messages whose length is smaller than or up to 8 packets can be transmitted within a few slots (7 at a maximum, beyond the slot in which the first is transmitted), data messages whose length exceeds 8 packets need to wait for more than a frame for the completion of their transmission. As a consequence of these two situations, the average time needed for the transmission of a data message after this message has been allocated its first slot is at least somewhat longer than half a frame (6 ms). Additionally, we see from figure 7 that the mean data message delay for RRA is maintained below 200 ms until the data message arrival rate λ equals about 2.5 messages/frame, then it increases sharply (about 400 ms for $\lambda = 2.55$ and off the scale for $\lambda = 2.6$).

In VDI-MCC, on the contrary, the mean data message delay is impressively lower than that in RRA for $\lambda < 2.6$, it remains below 200 ms for arrival rates up to 5.76 messages/frame before it starts to increase sharply and eventually goes off the scale for $\lambda = 5.82$. Consequently, in RRA the maximum data packet throughput achieved with the mean data message delay below 200 ms is about 20 (2.5×8) packets per frame, which corresponds to a 41.7% channel throughput[5] (i.e., 20/48). In VDI-MCC, in turn, the maximum data packet throughput achieved with the mean data message delay below 200 ms is 46.08 (5.76×8) packets per frame, which corresponds to a 94% channel throughput, *more than twice as much as the channel throughput of the RRA.*

4.1.3. Results for voice–data integrated access

Our simulation results, shown below, demonstrate that the use of only one slot for accommodating voice and data requests is sufficient and offers much better voice and data QoS than RRA. Before discussing our results, we support this argument analytically. The expected number of voice terminals entering talkspurt at the beginning of a frame is equal to the product of the mean number of silent terminals times the silence-to-talkspurt transition probability (i.e., $Np_S p_{ST}$). This expected number of new contenders per frame turns out to be much less than one (e.g., it varies between 0.34 and 0.38 for high load N values from 90 to 100). Because of this, collisions among VTs occur rarely. Therefore, in most cases no more than two minislots are needed to resolve voice contention, thus leaving 4 minislots for the transmission of data requests.

[5] Throughout this paper, we calculate the channel throughput as the used fraction of the total number of information slots within the frame.



(a)

(b)

Figure 8. Steady state mean data delays: (a) $N = 80$, (b) $N = 90$.

Figures 8(a) and (b) show the DaD and DmD curves for both schemes, for a constant number of VTs (80 in 8(a) and 90 in 8(b), which correspond to a medium and a high voice load, respectively) and for different data message arrival rates. We observe that the mean data message delay, for both medium and high voice loads, is in VDI-MCC not only much smaller than the mean data message delay in RRA, but also smaller than the mean data access delay in RRA (i.e., *in VDI-MCC data messages are transmitted faster than the transmission of just one data packet in RRA*). Furthermore, for VTs = 80 RRA achieves a channel throughput of 90.3% (for $\lambda = 1$) with the average data message delay below the limit of 200 ms. The respective channel throughput when VTs = 90 is 87.9% (for $\lambda = 0.3$). The corresponding channel throughput results for the VDI-MCC protocol are 91.8% (for $\lambda = 1.2$) and 91% (for $\lambda = 0.6$), respectively. Therefore, the channel throughput in our scheme is consistently greater than that of RRA.

The advantageous results of VDI-MCC are again owed to the immediate allocation of resources, to the use of ER slots, and to the exploitation of the frames where voice load happens (in spite of the considerable number of VTs in the system) to be lower than the set low frame voice occupancy limit of 95%. In the latter cases, the beneficiary for data users *low-voice-load mechanism* is activated. These three factors are responsible for the decrease of the mean data message delay in VDI-MCC (compared to that of RRA) by more than 80 ms (and up to 200 ms, see figure 8(b)), which corresponds to a constant improvement of at least 7 channel frames. This result is explained by both the quick transmission of data messages consisting of less than 8 packets (in low-voice-load situations their transmission takes place within one frame, offering an advantage of almost 7 frames to our scheme) and by the data preemption policy adopted in RRA, which furthermore aggravates the data delay performance of that scheme under medium and high voice load conditions.

Table 3 presents the results for the maximum voice capacity achieved by VDI-MCC for different data message arrival rates (in this case, we ignore the data QoS requirement for an upper bound on the mean data message delay).

As we have thoroughly explained in our system model, our scheme repeatedly offers absolute priority to voice users, in both the allocation of the available information slots and the acquisition of the ER slots within each frame. Thus, we consider unnecessary and unfair to data users the further imposition of a penalty in the form of data preemption in order to accommodate a newly arrived voice request, and hence, we do not adopt the approach used in RRA. As a result of this decision, we can see in table 3 that the increase of λ leads to a slight decrease of the voice capacity of our system. This decrease does not exceed a very small number of terminals (2 VTs). Still, VDI-MCC achieves a higher voice capacity than RRA, where the data preemption policy maintains the maximum voice capacity constant at 97 VTs, for all data rates considered.

When comparing the two schemes on the basis of the voice capacity and the channel throughput while at the same time fulfilling both the voice and data QoS requirements, it is clear from table 4 that our scheme is significantly better than RRA.

Table 3
First QoS comparison of VDI-MCC and RRA.

λ (messages/frame)	Max. voice capacity for $P_{\mathrm{drop}} < 1\%$ VDI-MCC	RRA
0.1	100	
0.2	99	
0.5	99	
0.7	99	97
0.8	98	
1.0	98	
1.2	98	
1.3	98	

4.2. VDI-HCC

The proper choice of the amount of the permanent request bandwidth, the use of the ER slots and the high capacity of the channel leads VDI-HCC to achieve a very high multiplexing gain. For voice traffic only, the maximum voice capacity reached is 556 terminals, which corresponds to a voice multiplexing gain of $556/256 = 2.17$. With the mean silence and talkspurt duration parameters of a voice terminal, the optimal (however, impossible to achieve in practice) multiplexing gain is equal to $\{(1.41 + 1.78)/1.41\} = 2.26$. Additionally, as expected from the very high multiplexing gain, the voice packet throughput is equally high, reaching 243.5 packets/frame at voice capacity, thus filling 97.4% of the information slots.

In figures 9 and 10 we present the simulation results for the voice packet dropping probability and the mean voice packet access delay versus the number of active voice terminals, respectively.

In figure 11 we consider the voice–data integration and we present the simulation results of our scheme for P_{drop} as a function of the number of VTs in the system, para-

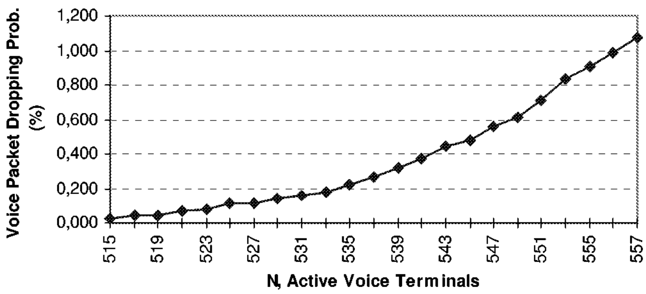


Figure 9. Steady state voice P_{drop} versus the number of active voice terminals.

Table 4
Second QoS comparison of VDI-MCC and RRA.

λ (messages/frame)	VDI-MCC Capacity	VDI-MCC Channel throughput	VDI-MCC P_{drop} (%)	VDI-MCC Voice packet access delay (ms)	VDI-MCC DmD (ms)	RRA Capacity	RRA Channel throughput	RRA P_{drop} (%)	RRA Voice packet access delay (ms)	RRA DmD (ms)
	Max. voice capacity for $P_{\mathrm{drop}} < 1\%$ and DmD < 200 ms									
0.1	100	0.909	0.973	23.14	180.6	95	0.886	0.623	26.69	197.4
0.2	98	0.912	0.761	19.62	196.7	92	0.877	0.361	22.99	191.3
0.3	96	0.909	0.526	15.93	183.1	91	0.886	0.280	21.84	194.6
0.4	94	0.910	0.417	14.22	194.7	89	0.885	0.187	20.49	189.4
0.5	92	0.909	0.282	11.94	177.9	88	0.892	0.153	20.04	199.4
0.6	91	0.918	0.236	11.22	199.2	86	0.891	0.102	19.28	193.3
0.7	89	0.917	0.176	10.05	187.3	85	0.899	0.084	19.03	198.7
0.8	87	0.915	0.105	8.64	165.4	83	0.898	0.063	18.69	192.2
0.9	86	0.922	0.082	8.36	186.1	82	0.905	0.048	18.48	198.4
1.0	85	0.928	0.077	7.98	192.4	81	0.912	0.042	18.38	199.5
1.1	83	0.926	0.049	7.29	182.2	79	0.911	0.039	18.28	190.4
1.2	81	0.927	0.032	6.81	175.3	78	0.918	0.036	18.24	197.3
1.3	80	0.933	0.031	6.71	195.6	76	0.916	0.032	18.14	188.3

metrized on the data message arrival rate λ. From the four curves shown in the figure, which, for the facts of the channel, correspond to low ($\lambda = 0.4$), medium ($\lambda = 1$, $\lambda = 2$) and high ($\lambda = 9$) data message arrival rates, we observe that the voice capacity does not drop below 552 VTs, even for $\lambda = 9$, which corresponds to a literal avalanche of data traffic (2.3 Mbps). The decrease of 4 VTs in the voice capacity of the system corresponds to just a 0.72% loss from the voice capacity achieved in the absence of data traffic. This result proves once more, similarly to the case of VDI-MCC, that the absence of a data preemption mechanism (such as the one introduced in RRA) leads to a trivial decrease in the performance of VDI-HCC on the voice performance and QoS parameters (notice that the voice capacity of 552 VTs corresponds to a very high multiplexing gain of $552/256 = 2.156$). This decrease in voice capacity is a small cost to afford for achieving a much better performance on the data message delay than the one we would have achieved had we used a data preemption mechanism, which severely deteriorates the data message delays under high voice loads.

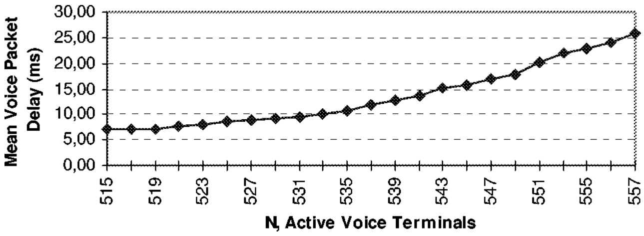


Figure 10. Steady state mean voice access delay versus the number of active voice terminals.

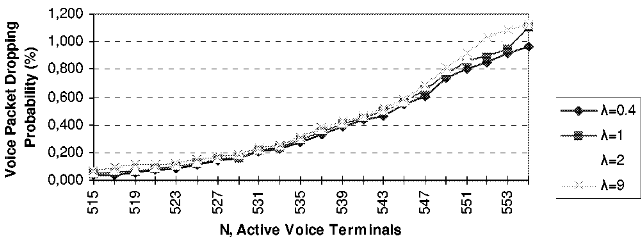


Figure 11. P_{drop} for different data message arrivals rates.

Figure 12 shows the DaD and DmD curves for VDI-HCC, for a constant number of VTs equal to 515 and for different data message arrival rates. We see that VDI-HCC achieves a very high aggregate channel throughput of 96.8% while the average data message delay remains below the delay limit of 200 ms (also notice from figure 11, that P_{drop} is much lower than 1% in this case). It is also shown that the mean data message delay is slightly higher than the mean data packet access delay, which proves that the low-voice-load mechanism guarantees the quickliest possible transmission to data messages after they have succeeded in transmitting their first packet, without affecting the overall throughput, voice performance and data packet access delay.

Finally, in table 5 we present the results for the voice capacity and the respective channel throughput achieved by VDI-HCC for different data message arrival rates, when at the same time both the voice and data QoS requirements are satisfied. The results concerning the channel throughput

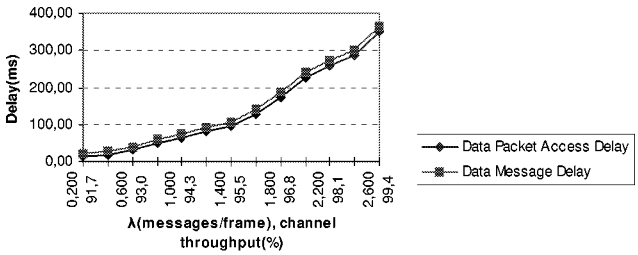


Figure 12. Steady state mean data delays with $N = 515$ active voice terminals.

Table 5
Performance of VDI-HCC when fulfilling both the QoS requirements.

λ (messages/frame)	Max. voice capacity for $P_{\text{drop}} < 1\%$ and DmD < 200 ms			
	VDI-HCC			
	Capacity	Channel throughput	P_{drop} (%)	DmD (ms)
0.2	544	0.963	0.469	195.9
0.4	538	0.960	0.292	198.6
0.6	535	0.961	0.242	189.3
0.8	530	0.962	0.184	192.4
1.0	525	0.960	0.139	187.8
1.5	520	0.966	0.098	193.0
2.0	511	0.966	0.035	182.3
2.5	505	0.972	0.029	187.4
3.0	500	0.977	0.021	188.3
4.0	486	0.982	0.015	199.6
5.0	469	0.985	0.012	196.5
6.0	448	0.981	0.007	189.9
7.0	427	0.980	0.006	193.3
8.0	409	0.979	0.005	198.2
9.0	391	0.979	0.003	193.6
10.0	372	0.977	0.003	187.2

are of major importance, because they prove that VDI-HCC takes advantage of all the available channel resources. The constant surpassing of 96% channel throughput for all the data message arrival rates and the achievement of a throughput as high as 98.5% for high data message arrival rates indicates the efficiency of the proposed multiplexing mechanism.

5. Conclusions

In this paper we have proposed and evaluated two multiple channel access control schemes for integrating voice and data traffic in both medium and high capacity microcellular environments. Voice traffic is offered almost absolute priority over the data traffic, due to its more stringent quality of service requirements.

Via an extensive simulation study we demonstrate that the scheme investigated for the medium capacity channel (corresponding mostly to outdoor environments) evidently excels both conceptually and in simulation results when compared to two other efficient existing MAC schemes, and that the scheme investigated for the high capacity channel (referring to indoor environments) achieves remarkably high channel throughput while fulfilling the voice and data quality of service and performance requirements. The very good results achieved by both our schemes are a consequence of the combination of two novel ideas of ours, along with two useful ideas presented in the literature in the near past (concept of ER slots and uniform distribution of request slots). Our two novel ideas are *the sharing of certain request slots among voice and data terminals with priority given to voice, and the use of a fully dynamic low-voice-load mechanism* in order to minimize unnecessary data message delays, reduce the voice packet dropping probability and hence increase the channel throughput by "exploiting" all the available information slots within each frame in the best possible way.

We should note that the high capacity channel offers the opportunity of broadening the classes of traffic it can accommodate. Therefore, the next step of our research will be the introduction of variable bit rate compressed video sources into the system and the efficient multiplexing of all three diverse types of traffic. This way, our scheme will serve to offer access to a complete multimedia platform.

References

[1] D. Bertsekas and R. Gallager, *Data Networks*, 2nd edition (Prentice-Hall, 1992).

[2] P.T. Brady, A technique for investigating on–off patterns of speech, Bell Systems Technical Journal (January 1965).

[3] A.C. Cleary, Voice–data channel access integration in third generation microcellular wireless networks: Design and performance evaluation, Ph.D. Dissertation, CIS Dept., University of Delaware (1997).

[4] A.C. Cleary and M. Paterakis, On the voice–data integration in third generation wireless access communication networks, European Transactions on Telecommunications and Related Technologies 5(1) (January–February 1994) 11–18.

[5] A.C. Cleary and M. Paterakis, An investigation of reservation random access algorithms for voice–data integration in microcellular wireless environments, International Journal of Wireless Information Networks 2(1) (January 1995) 1–16.

[6] A.C. Cleary and M. Paterakis, Design and performance evaluation of an RRA scheme for voice–data channel access in outdoor microcellular wireless environments, Mobile Networks and Applications (MONET) Journal 2(1) (1997) 31–43.

[7] D.C. Cox, Wireless personal communications: What is it?, IEEE Personal Communications 2(2) (April 1995) 20–35.

[8] J.M. De Vile, A reservation based multiple access scheme for a future universal mobile telecommunications system, in: *Proc. of the 7th IEE Conf. on Mobile and Personal Commun.*, Brighton, UK (December 1993) pp. 210–215.

[9] J. Dunlop, J. Irvine, D. Robertson and P. Cosimini, Performance of a statistically multiplexed access mechanism for a TDMA radio interface, IEEE Personal Communications 2(3) (June 1995) 56–64.

[10] D.A. Dyson and Z.J. Haas, A dynamic packet reservation multiple access scheme for wireless ATM, in: *Proceedings of the IEEE MILCOM'97*.

[11] A.M. Law and W.D. Kelton, *Simulation Modeling and Analysis*, 2nd edition (McGraw-Hill, 1991).

[12] N.M. Mitrou, G.L. Lyberopoulos and A.D. Panagopoulou, Voice and data integration in the air-interface of a microcellular mobile communication system, IEEE Transactions on Vehicular Technology 42(1) (February 1993) 1–13.

[13] N.M. Mitrou, Th.D. Orinos and E.N. Protonotarios, A reservation multiple access protocol for microcellular mobile-communication systems, IEEE Transactions on Vehicular Technology 39(4) (November 1990) 340–351.

[14] B. Mukherjee, Integrated voice–data communication over high speed fiber optic networks, IEEE Computer Magazine 24(2) (1991) 49–58.

[15] S. Nanda, D.J. Goodman and U. Timor, Performance of PRMA: A packet voice protocol for cellular systems, IEEE Transactions on Vehicular Technology 40 (1991) 584–598.

[16] M. Paterakis, L. Georgiadis and P. Papantoni-Kazakos, A full sensing window RA algorithm for networks with strict delay constraints, Algorithmica 4 (1989) 313–328.

[17] M. Paterakis and P. Papantoni-Kazakos, A simple window random access algorithm with advantageous properties, IEEE Transactions on Information Theory 35(5) (September 1989) 1124–1130.

[18] R. Rom and M. Sidi, *Multiple Access Protocols: Performance and Analysis* (Springer Verlag, 1990).

[19] T. Suda and T. Bradley, Packetized voice/data integrated transmission on a token passing ring local area network, IEEE Transactions on Communications 37(3) (1989) 238–244.

[20] W.C. Wong and D.J. Goodman, A packet reservation multiple access protocol for integrated speech and data transmission, IEE Proceedings-1 139 (December 1992) 607–612.

Polychronis Koutsakis received his Diploma degree in electrical engineering in 1997 from the University of Patras, Greece, and his M.Sc. degree in computer engineering in 1999 from the Technical University of Crete, Greece, where he is currently pursuing the Ph.D. degree in computer engineering. His research interests focus on wireless networks, with emphasis on multiple access, distributed protocols and queuing. He is a Student Member of the IEEE, and a member of the Greek Chamber of Professional Engineers, and the Greek Association of Electrical and Mechanical Engineers.
E-mail: polk@telecom.tuc.gr

Michael Paterakis received his Diploma degree from the National Technical University of Athens, Greece, his M.Sc. degree from the University of Connecticut, and his Ph.D. degree from the University of Virginia, in 1984, 1986, and 1988, respectively, all in electrical engineering. Since 1995, he is a Faculty Member in the Department of Electronic and Computer Engineering at the Technical University of Crete, Greece, where he is currently a Professor and Department Chair. He was an Associate Professor in the Department of Computer and Information Sciences at the University of Delaware, where he has been since September 1988. His research interests include computer communication networks with emphasis on protocol design, modeling and performance evaluation of broadband high speed networks, of multiple access wireless microcellular communication systems, and of packet radio networks; centralized and distributed multimedia information delivery systems; queuing and applied probability theory and their application to computer communication networks and information systems. He has published extensively in archival scientific journals, refereed conference proceedings and edited books, in the above-mentioned technical areas. He served on the Technical Program Committees of the 1991 International Conference on Distributed Computing Systems, the 1992 and 1994 IEEE INFOCOM Conferences, the 1997 IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems (HPCS'97), and the 6th IFIP Workshop on Performance Modeling and Evaluation of ATM Networks (IFIP ATM'98). Professor Paterakis is a Senior Member of the IEEE. He is also a member of the IEEE Technical Committee on Computer Communications, the Greek Chamber of Professional Engineers, and the Greek Association of Electrical and Mechanical Engineers.

E-mail: pateraki@telecom.tuc.gr