

A Near-Optimum MAC Protocol Based on the Distributed Queueing Random Access Protocol (DQRAP) for a CDMA Mobile Communication System

Luis Alonso, *Member, IEEE*, Ramón Agustí, *Member, IEEE*, and Oriol Sallent, *Member, IEEE*

Abstract—This paper presents and analyzes a new near-optimum medium access control (MAC) protocol. The proposed access scheme is suitable for a CDMA mobile communication environment, and keeps under control and upper bounded the number of simultaneous transmissions. It has a delay performance approaching that of an ideal optimum M/M/K system, where K is the number of spreading codes being used (maximum number of simultaneous transmissions). The protocol is a free random access protocol when the traffic load is light, and switches smoothly and automatically to a reservation protocol when traffic load becomes heavier. It is based on distributed queues and a collision resolution algorithm. Moreover, a physical receiver structure is proposed and analyzed in order to preserve the robustness of the protocol in a wireless link. The results obtained show that the protocol outperforms other well known medium access protocols in terms of stability and delay, even when taking into account the loss caused by channel propagation conditions.

Index Terms—Code division multiaccess, mobile communications, multiaccess communication, protocols.

I. INTRODUCTION

IN THE LAST few years, many research efforts have focused on the design of medium access control (MAC) protocols. In the future third-generation communication systems, mixed services and different traffic patterns will have to share the same channel structure and resources. MAC techniques must provide flexibility and efficiency to allow the existence of these types of systems with reasonable complexity and reliability.

ALOHA and slotted-ALOHA techniques have been widely used in the past as random access protocols. However, their low throughput (0.18 and 0.36 maximum) and potential instability at heavy traffic load have led to the appearance of collision resolution algorithms (CRA), also called tree algorithms [1], which have a higher performance (up to 0.568 based on ternary channel feedback [2]). Some protocols achieve higher throughput by using control minislots for reservation purposes. Of all these, the announced arrival random access protocols

(AARA) [3] achieve the best delay and throughput performance (0.853 with only three control minislots). However, to reach throughputs approaching unity, the AARA protocols need a theoretically infinite number of minislots, and this is obviously impractical and inefficient because of the overhead introduced by each minislot.

One widely studied medium access protocol based on control minislots is DQRUMA (distributed queue request update multiple access) [15]. This protocol uses a certain number of access minislots for reservation purposes. Terminals with data to transmit send an access request in one of these minislots applying a slotted-ALOHA strategy. This request contains the identification number of the terminal and the type and quality of the demanded service. The main advantage of using this centralized strategy is that it allows the designer to totally control the behavior of the system. It is possible to give priority to terminals with strict quality requirements, such as tight delay bounds, instead of simply maximizing the overall throughput. However, high complexity algorithms, a great amount of signaling and feedback information, and accurate admission control policies are required for the system to work correctly. Moreover, slotted-ALOHA strategy is used for accessing purposes, and thus the potential instability problem is still present when traffic load is high.

In general, merely using control minislots makes the system more complex as it is necessary to have time slots with different time sizes. Nevertheless, we observe that all existing tree protocols that do not have minislots use data slots to resolve collisions, and thus lose the channel capacity of all the empty slots or collided packets. The suggested improvements to tree protocols seek to reduce the number of collisions and empty slots, but they do not eliminate this type of efficiency loss. Keeping all these ideas in mind, Xu and Campbell proposed the distributed queueing random access protocol (DQRAP) [4], [5], [19], which seems to be one of the best-performing MAC protocols proposed to date. This protocol uses three control minislots and is based on a tree-type collision resolution algorithm. It was initially designed for a TDMA environment, particularly for the distribution of CATV (cable TV) signal. Inspired by DQDB (distributed queueing dual bus, now the IEEE 802.6 standard for metropolitan area networks), its performance approaches that of an ideal M/D/1 queue, reaching maximum stable throughputs close to one, and maintaining its stability for traffic loads up to

Manuscript received July 1, 1999; revised February 25, 2000. This work was supported by CYCIT Project TIC 98-684. Part of this work was presented at PIMRC'99, Osaka, Sept. 1999; and at VTC'99 Fall, Amsterdam, Sept. 1999.

The authors are with the Department of Signal Theory and Communications, Universitat Politècnica de Catalunya (UPC), Barcelona 08034, Spain (e-mail: luisg@tsc.upc.es; ramon@tsc.upc.es; oriol@tsc.upc.es).

Publisher Item Identifier S 0733-8716(00)07135-3.

channel capacity. These near-optimum characteristics add to the appeal of using the rationale of this protocol in other transmission environments such as packet radio systems.

On the other hand, direct-sequence code division multiple access (DS-CDMA) is going to be adopted for third-generation mobile telecommunication systems. Schemes based on wide-band CDMA (WCDMA) [6] have been chosen as radio interfaces by the standardization body in Japan (ARIB), and also in Europe by the ETSI for the UMTS Terrestrial Radio Access (UTRA) [7]. This access scheme is also being considered in the International Mobile Telecommunication 2000 (IMT-2000) [8] by the ITU. In this paper, we propose a random near-optimum medium access protocol that modifies and extends DQRAP techniques for use in a CDMA environment such as those mentioned above. The operation mode of the protocol may allow the use of random access channels (RACH) or other packet transmission systems, in uplinks (reverse links), not only for accessing purposes but also to efficiently transmit data.

For this purpose, the idea of using a DQRAP engine for each one of the spreading codes is introduced. Then, as the protocol is based on two logical distributed queues (the collision resolution queue and the transmission queue), the queues corresponding to each spreading code are joined in only one queue for each group (resolution and transmission). We will show in this paper that the DQRAP/CDMA protocol can be modeled as two concatenated M/M/K systems, where K is the number of available spreading codes. Moreover, DQRAP/CDMA is provided with a mechanism that reduces to a minimum the *jitter* in the delay of the packets corresponding to one message, and also becomes a new advantage for managing messages of more than one slot length.

The protocol is a free random access protocol when the traffic load is light, thus reducing the transmission delay, and switches smoothly and automatically to a reservation protocol when traffic load becomes heavier, blocking the transmission of newly arrived packets by putting them into a data transmission queue. Then, given certain CDMA channel characteristics (i.e., spreading factor, bits per slot, fading and interference model, diversity, coding, ARQ strategy, etc.), DQRAP/CDMA allows an optimum number of simultaneous transmissions to be kept in the system, avoiding collisions to a great extent (they only could appear for light traffic conditions) and preventing the use of more receiver resources than strictly needed. This behavior is the key to its good delay and throughput performance.

In order to assess the DQRAP/CDMA scheme under realistic conditions, a receiver scheme for the control minislots detection was proposed and analyzed. Expressions for the minislots state misdetection probabilities were derived, and various mechanisms were introduced to keep the robustness of the protocol in a Rayleigh fading channel situation. Finally, a comparison was made to other MAC schemes extensively studied in the open literature such as slotted-ALOHA/CDMA [9] and ISMA/CDMA [14]. The results obtained show a significant improvement in the system delay and throughput performance.

The paper is organized as follows. The protocol description is detailed in Section II. In Section III, the analytical model is presented and studied. Expressions for the total system delay are

also derived in this section. Section IV explains and analyzes the proposed scheme for the control minislots state detection. In this section, protocol algorithm modifications are also introduced to recover from errors in the minislots detection. Section V shows computer simulation results and comparisons to other protocols. Finally, Appendix I and Appendix II are devoted to the conclusion.

II. PROTOCOL DESCRIPTION

Let us consider N data terminals which share a CDMA channel with K available spreading codes to communicate with a base station. The time axis is divided into slots, and each slot has two fields. The first field is the access field, which is further divided into m control minislots. The second field is the data part, where terminals will transmit their packets. We assume that every station has perfect slot and minislots synchronization. The K spreading codes are put in order, and we will denote K_i for the i th code. We consider that the terminals are able to change the spreading code for data and request transmission on a slot-by-slot basis. The messages generated by one terminal are split into slot-duration packets and put into a buffer. Each packet will be sent with the same spreading code, but not all the packets pertaining to one message will necessarily be sent with the same spreading code.

The protocol uses two concatenated distributed queues: the collision resolution queue and the data transmission queue. When a message arrives at the system, the corresponding terminal, following a certain set of rules described below, selects a spreading code and sends a request in one of the control minislots pertaining to this code. If it fails (i.e., the request collides with one or more requests from other messages), it enters the collision resolution queue. Collisions are then resolved in the order fixed by the queue discipline. In addition, the data transmission queue contains the messages that have succeeded in their request and are waiting to be transmitted to the base station also following the order fixed by the corresponding queue discipline. Collision resolution and data transmission processes work in parallel.

All the terminals must have four integer counters, which represent the two logical distributed queues. We will denote them as TQ, RQ, pTQ, and pRQ. TQ is the number of messages waiting for transmission in the distributed transmission queue. RQ is the number of collisions waiting for resolution in the distributed collision resolution queue. pTQ is the position of a given terminal in the data transmission queue, and pRQ is the position of that terminal in the collision resolution queue. These values range from 0, meaning that the terminal does not have any position in the corresponding queue, to TQ or RQ (respectively), 1 being the first position of the queue. TQ and RQ have the same value for all the terminals in the system (i.e., they represent *distributed* queues), while pTQ and pRQ have a specific value for each terminal. We assume both queues to be FIFO. All four values are initially set to zero and must be kept updated using the feedback information sent by the base station, each slot, using a broadcast channel, and following a set of rules described below. It consists of ternary state data for each control minislots of every spreading code, and also has to include a final-message-bit for each code.

The three different states that the base station must be able to distinguish are: empty, success, and collision. A collision will occur when more than one station transmits in the same minislot of the same spreading code. The final-message-bit is the mark that all the data terminals must send when they are transmitting the last packet from one message. This flag bit must be ON in the last packet of each message, and must be OFF in all the other packets. This mechanism allows all packets from a message to be transmitted with a single request and minimizes the delay *jitter* between these packets. Nevertheless, if propagation delay in the system prevents the terminals from receiving the feedback information about this final-message-bit before the next data slot begins, an empty slot loss is produced at the end of each message. If messages are known to be short (for example, ATM cells), it should be possible and convenient to switch off this mechanism and consider all messages formed by a single packet.

The protocol algorithm consists of three sets of rules that each data terminal has to follow at the end of each slot. They are, in order of execution, the queueing discipline rules (QDR), the data transmission rules (DTR), and the request transmission rules (RTR).

A. Algorithm Rules

We will now describe the algorithm rules that each data terminal has to execute at the end of each slot, assuming that, at this time, the feedback information from the base station about the state of the control minislots of the previous slot has already been received by the terminal. They must be executed in the order presented below. Some rules have initial conditions that must be true to execute the corresponding actions. If the assertion is not verified, then the algorithm simply jumps to the next rule. When all the rules have been checked, the slot finishes and a new one starts.

1) QDR (Queueing Discipline Rules):

- a) Each station increments the value of TQ by one unit for each control minislot in the success state, taking into account the feedback information from all the control minislots from any of the K spreading codes.
- b) Each station reduces the value of TQ by one unit for each packet correctly received by the base station with the final-message-bit set to ON from any of the spreading codes.
- c) If $RQ > 0$, each station reduces the value of RQ by $\min(RQ, K)$ units.
- d) Each station increments the value of RQ by one unit for each control minislot in the collision state, taking into account all the control minislots from any of the K spreading codes.
- e) Depending on its state, and the results of the control minislots, each station calculates the values for pTQ and pRQ. That is, if it has sent a request and this request has succeeded, it calculates its position among all the succeeding minislots and sets pTQ to the corresponding value at the end of TQ. For this purpose, all the successes are sorted using the order of the spreading code to which they belong, and within the same spreading code, using a time

arrival criterion. On the other hand, if the request has collided, the terminal calculates its position among all the present collisions and sets pRQ to the corresponding value at the end of RQ. If it has not sent any request, then pTQ and pRQ follow the same update rules as TQ and RQ, respectively, but only if the initial values are other than zero.

2) DTR (Data Transmission Rules):

- a) If $TQ < K$, each station that has $pTQ = 0$, $pRQ = 0$ and data packets ready to be sent transmits the first packet of its buffer using the spreading code K_{TQ+1} . This rule is also called the free access rule, as it allows newly arrived packets to be transmitted immediately when traffic load is light. However, using this rule may cause a collision in the data part of a slot.
- b) If a station has $pTQ > 0$ and $pTQ \leq K$, the station transmits the first packet of its buffer using the spreading code K_{pTQ} . If this packet is the last one of the current message, the station sets the final-message-bit to ON.

3) RTR (Request Transmission Rules):

- a) If $RQ < K$, each station that has $pRQ = 0$ and $pTQ = 0$ and data packets ready to be sent randomly selects one of the control minislots of the spreading code K_{RQ+1} and transmits a request in it.
- b) If a station has $pRQ > 0$ and $pRQ \leq K$, the station randomly selects one of the control minislots of the spreading code K_{pRQ} and transmits a request in it.

B. Example

The example shown in Fig. 1 illustrates the operation of the protocol with $K = 3$, $N = 4$, and $m = 3$ starting from an idle system (all values are initially zero). All the messages generated by the terminals are assumed to be of length one, so each data slot has the final-message-bit set to ON.

In slot $n = 0$, three messages arrive at the system. In $n = 1$, they try to send a request and also to transmit the data in the first spreading code (using rules RTR-1 and DTR-1). Only the request from $p2$ succeeds and enables $p2$ to enter the transmission queue. As the requests of $p1$ and $p3$ collide, they enter the collision resolution queue. All packets use the free access rule (DTR-1), and then the data part also collides. In this slot, a message from $p4$ arrives at the system.

In $n = 2$, $p2$ is the only packet in the transmission queue and it is thus transmitted using the first spreading code (DTR-2). Packets $p1$ and $p3$ resolve their collision (RTR-2) and enter the transmission queue ($p1$ in the first position, as its request used a prior control minislot) (QDR-5). However, $p4$ transmits its request and data using the second spreading code (RTR-1 and DTR-1). As $p4$ is the only new packet arriving at the system, its data transmission succeeds, and therefore it does not need to enter any queue. Two more packets arrive at this slot.

In $n = 3$, $p1$ and $p3$ are transmitted using the first and second spreading codes (DTR-2). The new packets $p2$ and $p4$ send their requests and collide. They enter the collision resolution queue. In $n = 4$, requests from $p2$ and $p4$ collide and the packets again enter the collision resolution queue. The requests from $p1$ and

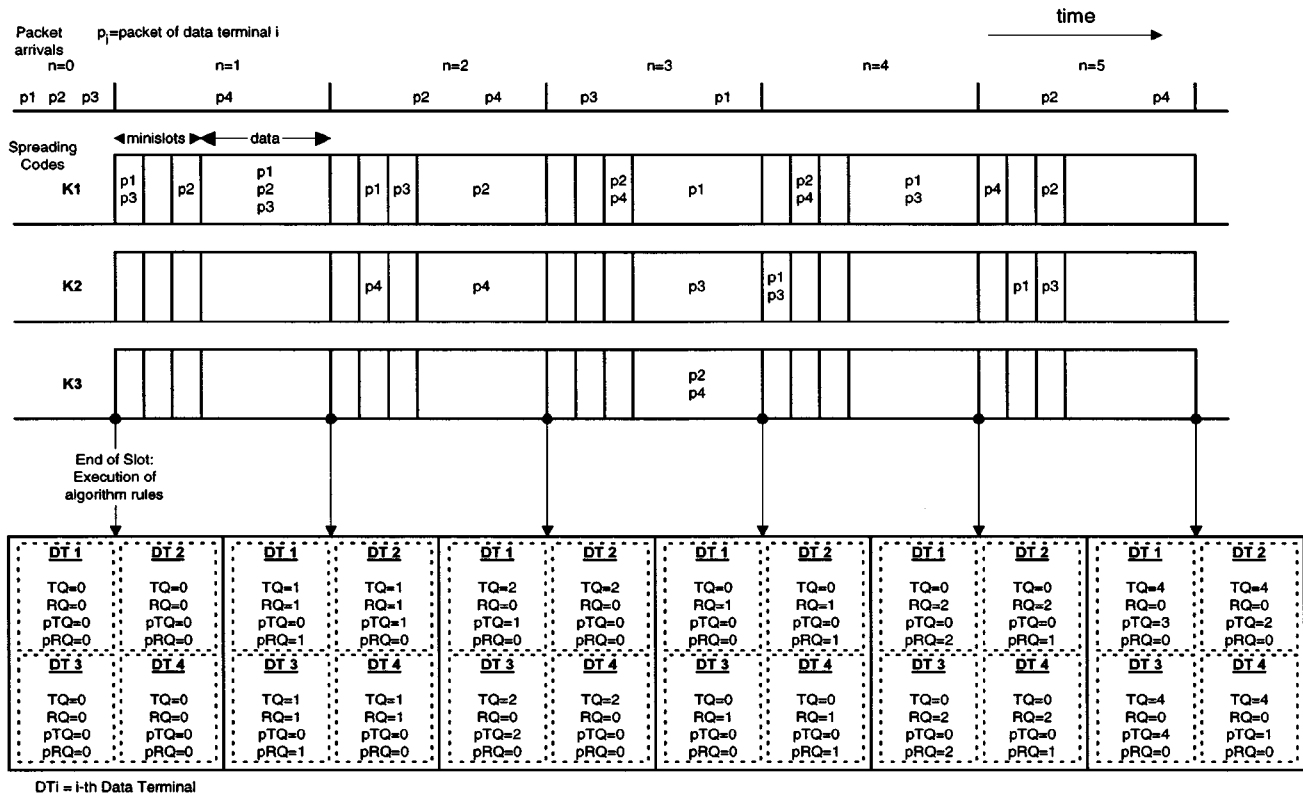


Fig. 1. Example of DQRAP/CDMA protocol operation.

p_3 also collide and enter this latter queue in the next position, as they have used a higher-in-order spreading code. In $n = 5$, all the packets attempt to resolve their collisions and succeed, entering the transmission queue. This process continues endlessly.

C. Practical Considerations

At this point, we are going to outline some practical considerations for real implementation purposes. First of all, we are considering that the feedback information about the states of the control minislots is broadcasted to the terminals and arrives before the next time slot begins. This is a feasible feature as the system has the data slot duration for performing this transmission.

Moreover, it is assumed that the base station also broadcasts the values of TQ and RQ periodically in the control downlink, in order to allow new users to join the system and recover from possible losses of the counters. This information consists of only two integer values that occupy a few bits. Another practical possibility is to transmit this number to the mobile terminals only when needed to join the system or recover from errors.

III. PROTOCOL MODEL AND ANALYSIS

The DQRAP/CDMA protocol can be modeled as shown in Fig. 2. We have two queue subsystems: the collision resolution subsystem and the transmission subsystem. The enable transmission interval (ETI) service time represents the time each message has to wait from when it arrives at the system until the next time slot starts. Normalizing the time axis in slot units, this

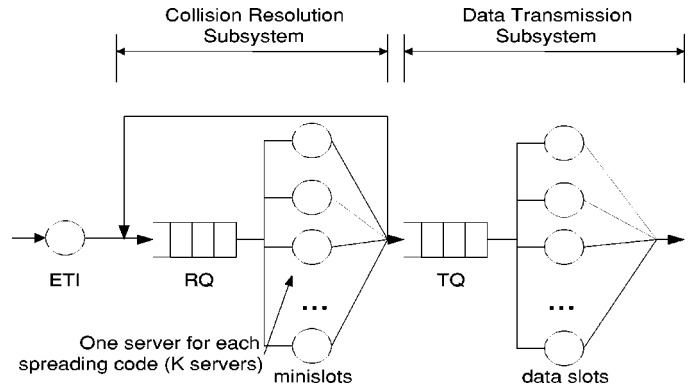


Fig. 2. Model of DQRAP/CDMA protocol.

service time will thus be a uniformly distributed random variable in the interval (0, 1). Both subsystems have as many servers as available spreading codes (i.e., K).

The elements in the system are the messages generated by the users, although they only use the control minislots for accessing purposes in the collision resolution subsystem. The feedback line in this subsystem represents that the messages that collide in their requests must enter the queue again until they succeed.

A. Delay Analysis

The total delay for a message (t_T) can be broken down into four terms: the service time of the ETI (t_{ETI}), the total delay of the collision resolution subsystem (t_{RQ}), the total delay of the data transmission subsystem (t_{TQ}), and the delay caused by the collision of a data packet in a data slot (t_c). This latter term appears when more than one terminal transmits its packet using

rule 1 of the DTR (the free access rule) in the same slot. Thus, the expected value of the total delay of the system is

$$E[t_T] = E[t_{ETI}] + E[t_{RQ}] + E[t_c] + E[t_{TQ}]. \quad (1)$$

We will now describe the expression of the terms in (1). First of all, $E[t_{ETI}]$ equals 0.5 because the arrival of messages is independent of the slot timing and, as noted above, t_{ETI} is a uniformly distributed random variable in the interval (0, 1).

1) *Total Delay of the Collision Resolution Subsystem*: Let $P(\lambda)$ be the probability that a message will find a free control minislot to access when it arrives at the system, where λ is the total message input rate to the system (with Poisson distribution). We may note that, according to RTR, all newly arrived messages use the same spreading code to send their request. In addition, the arrival process is memoryless, and the protocol uses a tree algorithm for collision resolution, that is, all packets that have collided in a certain minislot use an exclusive code to resolve their contention. Then, if we have m control minislots per code, it results in

$$P(\lambda) = \sum_{n=0}^{\infty} P(\text{free minislot} | k = n) P(k = n) \quad (2)$$

where $P(\text{free minislot} | k = n)$ is the probability of randomly choosing an empty minislot when n packets have arrived at the system in a given slot, and $P(k = n)$ is the probability that n packets arrive at the system in that slot. Therefore, it can be written that

$$\begin{aligned} P(\lambda) &= P(k = 0) + \sum_{n=1}^{\infty} P(k = n) m \left(\frac{1}{m}\right) \left(1 - \frac{1}{m}\right)^n \\ &= e^{-\lambda} + \sum_{n=1}^{\infty} \frac{\lambda^n}{n!} e^{-\lambda} \left(1 - \frac{1}{m}\right)^n \\ &= e^{-(\lambda/m)}. \end{aligned} \quad (3)$$

All the messages in the collision resolution subsystem (including both the messages waiting in the queue and the newly arrived ones) have a probability $P(\lambda)$ of succeeding in their request. Thus, the service time for the collision resolution subsystem will be a geometrically distributed discrete random variable $[t]$ (where $[\cdot]$ denotes the integer part), with probability distribution function (PDF):

$$F_{RQ}([t]) = 1 - (1 - P(\lambda))^{[t]}. \quad (4)$$

At this point, if we use the exact discrete service time distribution, the system is an M/G/K and, as pointed out in [10], this type of system is analytically unmanageable and only loose bound expressions exist for them. However, in our case, we can approximate the geometrical distribution by the corresponding exponential distribution for a continuous service time t as, in fact, the geometrical distribution values are only the sampling of the exponential one. Computer simulation results, as will be shown in Section IV, will confirm that this approximation is fea-

sible, as they fit this model very well. Then, with this assumption, we can write its probability density function *pdf* as

$$f_{RQ}(t) = \frac{\partial F_{RQ}(t)}{\partial t} = (1 - P(\lambda))^t \ln\left(\frac{1}{1 - P(\lambda)}\right). \quad (5)$$

We can thus see that the service time of the collision resolution subsystem is a Poisson-distributed random variable with mean

$$\frac{1}{\mu_{RQ}} = \left[\ln\left(\frac{1}{1 - P(\lambda)}\right) \right]^{-1}. \quad (6)$$

We can therefore model the system as an M/M/K. Following the analysis in [10], adding the waiting time in the queue plus the service time, we can write the total delay for the collision resolution subsystem

$$\begin{aligned} E[t_{RQ}] &= \frac{1}{\ln\left(\frac{1}{1 - e^{-\lambda/m}}\right)} \\ &\quad + \frac{P_{KRQ}}{K \ln\left(\frac{1}{1 - e^{-\lambda/m}}\right) (1 - \rho_{RQ})} \end{aligned} \quad (7)$$

where

$$\rho_{RQ} = \frac{\lambda}{K \mu_{RQ}} \quad (8)$$

and

$$P_{KRQ} = \frac{\frac{(K \rho_{RQ})^K}{K!(1 - \rho_{RQ})}}{\sum_{n=0}^{K-1} \frac{(K \rho_{RQ})^n}{n!} + \frac{(K \rho_{RQ})^K}{K!(1 - \rho_{RQ})}}. \quad (9)$$

This last expression is the Erlang C formula for the delay probability.

2) *Total Delay of the Data Transmission Subsystem*: As both arrival and service time processes are Poisson-distributed, the collision resolution subsystem output traffic pattern will also be Poisson-distributed and, as shown in [11], with the same rate as the input traffic λ . This output traffic is directly the input traffic of the data transmission subsystem.

All the terminals generate messages of exponentially distributed length with mean $(1/\mu)$. Then, assuming that the system uses a Stop & Wait ARQ strategy to retransmit each packet containing one or more error bits, the service time of the data transmission subsystem will also be exponentially distributed. The mean value of this service time will be the mean length of the messages, $(1/\mu)$, multiplied by the mean transmission time for each packet of the message, T_1 . Calling p the probability that a packet has at least one error bit, we can write the value for T_1 as

$$T_1 = (1 - p) \sum_{n=1}^{\infty} n p^{n-1} = \frac{1}{1 - p}. \quad (10)$$

If we discard and retransmit any packet having at least one erroneous bit, p is the block error ratio, *BLER*, so we can finally

write the mean service time for the data transmission subsystem ($1/\mu_{\text{TQ}}$) as

$$\frac{1}{\mu_{\text{TQ}}} = \frac{1}{\mu} T_1 = \frac{1}{\mu(1 - \text{BLER})}. \quad (11)$$

Again, both the input traffic and the service time of the data transmission subsystem are exponentially distributed, and we can thus model this subsystem as an M/M/K queue system. Its total delay expression thus has the same terms as the one for the collision resolution subsystem but changing the service time rate μ_{RQ} by the new value μ_{TQ} , that is,

$$E[t_{\text{TQ}}] = \frac{1}{\mu(1 - \text{BLER})} + \frac{P_{K\text{TQ}}}{K\mu(1 - \text{BLER})(1 - \rho_{\text{TQ}})} \quad (12)$$

where the expression for ρ_{TQ} is also the same as for ρ_{RQ} but substituting the value of μ_{RQ} with the new value μ_{TQ} . That is,

$$\rho_{\text{TQ}} = \frac{\lambda}{K\mu_{\text{TQ}}}. \quad (13)$$

Note that $P_{K\text{TQ}}$ is the Erlang C formula with K servers and with this new value ρ_{TQ} .

3) *Data Collision Delay*: According to the algorithm rules of the protocol, the only possible situation where a data collision can occur is when the system has fewer than K messages waiting in the data transmission subsystem, and more than one packet arrives at the system in the same slot. The mean delay caused by this event will be its probability, since if a data collision occurs, the message will enter any of the two subsystems of the model (depending on whether its request has succeeded or collided) and will no longer collide. We can evaluate this probability as

$$\begin{aligned} E[t_c] &= \left(\sum_{n=0}^{K-1} p_n \right) \left(\sum_{k=2}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \right) \\ &= \frac{\sum_{n=0}^{K-1} \frac{(K\rho_{\text{TQ}})^n}{n!}}{\sum_{n=0}^{K-1} \frac{(K\rho_{\text{TQ}})^n}{n!} + \frac{(K\rho_{\text{TQ}})^K}{K!(1 - \rho_{\text{TQ}})}} (1 - e^{-\lambda(1 + \lambda)}) \end{aligned} \quad (14)$$

where p_n is the probability that the system has n units, taking into account the ones in the queue and the ones being served.

4) *Total System Delay*: The average total delay for a message will be

$$E[t_T] = \frac{1}{2} + \left[\frac{1}{\ln\left(\frac{1}{1 - e^{-\lambda/m}}\right)} + \frac{P_{K\text{RQ}}}{K \ln\left(\frac{1}{1 - e^{-\lambda/m}}\right) (1 - \rho_{\text{RQ}})} \right]$$

$$\begin{aligned} &+ \left[\frac{1}{\mu(1 - \text{BLER})} + \frac{P_{K\text{TQ}}}{K\mu(1 - \text{BLER})(1 - \rho_{\text{TQ}})} \right] \\ &+ \frac{\sum_{n=0}^{K-1} \frac{(K\rho_{\text{TQ}})^n}{n!}}{\sum_{n=0}^{K-1} \frac{(K\rho_{\text{TQ}})^n}{n!} + \frac{(K\rho_{\text{TQ}})^K}{K!(1 - \rho_{\text{TQ}})}} (1 - e^{-\lambda(1 + \lambda)}). \end{aligned}$$

To evaluate this expression, we need to know the value of the BLER. If we assume a perfect power control for a steady state and neglect the effect of thermal noise, we may use the Gaussian hypothesis for the interferences. Then, as we discard and retransmit all the packets containing one or more errors, we can write [12]

$$\text{BLER} = 1 - \left(1 - \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{3S_f}{2(k-1)}} \right) \right)^L \quad (15)$$

where

- S_f spreading factor;
- k number of simultaneous data transmissions;
- L number of bits contained in the packets sent during a time slot.

Note that k is not constant with time. For analytical evaluation purposes, we will use $k = K$, as this is the worst case value.

However, this perfect power control is not available in the initial transient state when minislots are used to access the media. Therefore, a Rayleigh fading model can approach the communication channel better, as is shown in the following.

B. Detection of Access Requests in Control Minislots

One of the main problems for the practical implementation of protocols using minislots for accessing purposes is the complexity they entail in the physical layer. In normal conditions, the only difference between these control minislots and the data slots is their length, measured in bits or in time units. Unfortunately, regardless of the actual length of a slot, special symbols such as bit training patterns must be transmitted at the beginning of each slot for channel synchronization, equalization, and power control. The number of these symbols required depends on the characteristics of the radio link. The performance improvement of the minislots is thus impaired when taking into account this physical layer overhead. Moreover, mixed slot sizes complicate the hardware design of the radio interface.

However, DQRAP/CDMA has a critical advantage for tackling this problem. Control minislots are simply a burst of chips that a terminal has to send inside a certain window of time for the base station to detect its access demand. The only requirement is that it must be possible for the base station to distinguish between three different states: 1) empty, that is, no energy is received; 2) success, that is, a single burst from *any* terminal has been detected; and 3) collision, when two *or more* bursts have been detected.

The receiver structure for this access scheme could be as follows: each station has two different assigned access sequences, and no other terminal will have the same pair of sequences. When a terminal has to transmit an access burst in a control minislot, it will send both sequences simultaneously. The detection

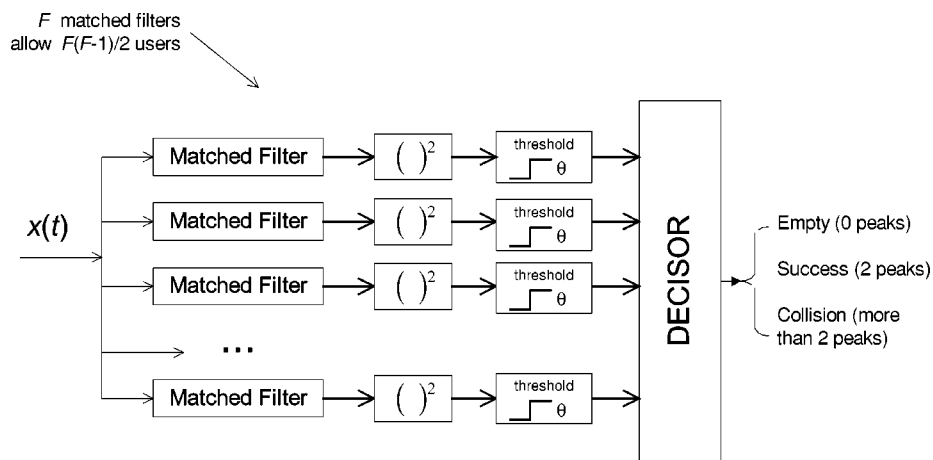


Fig. 3. Structure of the minislot receiver at the base station.

of more than two access sequences will allow the base station to detect collisions without any need to have one matched filter for each user. Fig. 3 shows the structure of the receiver at the base station. This receiver consists of a bank of matched filters, one for each different sequence. A matched filter will output a peak whenever it detects that any terminal has transmitted the corresponding sequence. Then, the decision block only needs to count the number of correlation peaks at the output of the bank of filters. Ideally, if two peaks are detected, it means that only one terminal has sent its request. A greater number of peaks will denote the presence of a collision. The absence of peaks simply reveals the absence of access requests. Note that if we use a bank of F filters, we can address $F(F-1)/2$ different users.

In order to assess the performance of the proposed receiver scheme in terms of the probability of minislot state misdetection, we first must calculate the detection and false alarm probabilities at the output of each detection filter, that is, the matched filter with the square power and threshold decision blocks.

C. Analysis of the Minislot State Detection Scheme

Using an optimal receiver scheme, with antenna and postdetection diversity of order D (see Appendix II), the false alarm probability P_f and the detection probability P_d for each detection filter are given by

$$P_f = \int_{\theta}^{\infty} p_0(Z) dZ = e^{-\theta/V} \sum_{n=0}^{D-1} \frac{(\theta/V)^n}{n!} \quad (16)$$

$$P_d = \int_{\theta}^{\infty} p_1(Z) dZ = e^{-\theta/V_F} \sum_{n=0}^{D-1} \frac{(\theta/V_F)^n}{n!} \quad (17)$$

where θ is the decision threshold, and

$$V = MI_0 = M(N_0 + \frac{2}{3} E_c(r-1)) \quad (18)$$

and

$$V_F = M(I_0 + ME_c). \quad (19)$$

M is the number of chips, I_0 is the total interference level, E_c is the energy per chip, and r is the number of simultaneous access request transmissions.

Note that V and V_F depend on the number of simultaneous user transmissions that cause interference in the system, that is, on the value of I_0 . In our practical case, the transmitted sequences are used for access request detection (and possible collisions) in the control minislots. This number of simultaneous transmissions thus matches the number of access request sequences sent in the same considered minislot, using any of the available spreading codes. Moreover, this value will depend on the traffic load offered to the system, measured in terms of the number of messages trying to access the channel per time unit (doubled). We must choose a value for r (the number of simultaneous access requests), which we will call design r , or simply r_D , and select the false alarm probability we wish for this specific value. Indeed, if we neglect the effect of the thermal noise (interference limited system), and denoting as P_{fD} the design false alarm probability, the value for the decision threshold can be explicitly written for $D = 1$ (no diversity)

$$\theta = -\frac{2}{3}(r_D - 1)ME_c \ln P_{fD}. \quad (20)$$

Therefore, the detection probability is

$$P_d = \exp\left(\frac{\frac{2}{3}(r_D - 1)ME_c \ln P_{fD}}{N(\frac{2}{3}(r_D - 1)E_c + ME_c)}\right) = P_{fD}^{(2(r_D-1)/2(r_D-1)+3M)}. \quad (21)$$

Fig. 4 shows the values for the false alarm and detection probabilities as a function of the parameter r_D for a sequence of length $M = 256$.

However, the real false alarm and detection probabilities in the system will not be as presented in this figure. Indeed, once the threshold θ for the decisor has been chosen, these probabilities still depend on the number of simultaneous access requests transmitted in each minislot, that is, the total interference level, which will not always be the design one r_D . In general, we will actually have a certain value for r different from r_D .

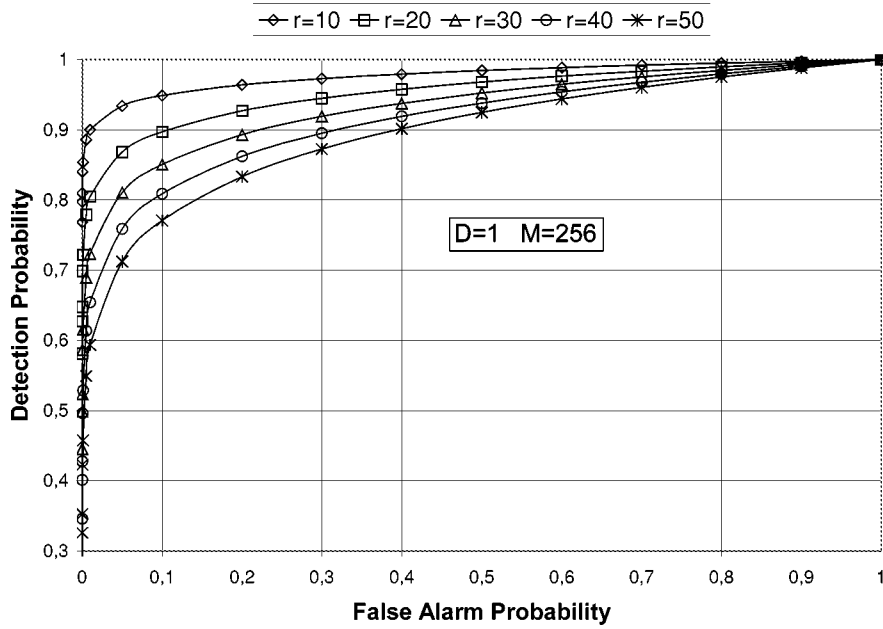


Fig. 4. Detection and false alarm probabilities in a Rayleigh fading environment without diversity.

Given the threshold value defined in (20), the expressions for the probabilities in the system will be

$$\begin{aligned}
 P_f &= e^{-\theta/V} \\
 &= \exp\left(\frac{\frac{2}{3}(r_D - 1)ME_c \ln P_{fD}}{\frac{2}{3}(r - 1)ME_c}\right) = P_{fD}^{(r_D - 1)/(r - 1)} \quad (22) \\
 P_d &= e^{-\theta/V_F} \\
 &= \exp\left(\frac{\frac{2}{3}(r_D - 1)ME_c \ln P_{fD}}{\frac{2}{3}(r - 1)ME_c + M^2 E_c}\right) \\
 &= P_{fD}^{(2(r_D - 1))/(2(r - 1) + 3M)} \quad (23)
 \end{aligned}$$

where r is the actual number of access requests. As an example, Figs. 5 and 6 show the false alarm and detection probabilities as a function of r for a $P_{fD} = 10^{-3}$ and with $r_D = 20$.

We can observe that the variation of the detection probability is very small with r . Furthermore, the false alarm probability also increases smoothly for $r > r_D$ and decreases abruptly when $r < r_D$. These properties match our practical application well: when traffic load is higher than the design rate, the probabilities are only slightly worse than decided; and when traffic load becomes lighter, the false alarm probability decreases dramatically, improving the system performance.

Using antenna diversity ($D > 1$), it is not possible to write the explicit expression of the threshold as a function of the false alarm probability. Let us use γ for the (θ/V) relation that fulfills

$$P_{fD} = e^{-\theta/V} \sum_{n=0}^{D-1} \frac{(\theta/V)^n}{n!} = e^{-\gamma} \sum_{n=0}^{D-1} \frac{\gamma^n}{n!} \quad (24)$$

where, again,

- P_{fD} design false alarm probability;
- r_D number of simultaneous access requests used for design;
- r number of actual simultaneous access requests.

The false alarm and detection probabilities are thus

$$P_f = e^{-(\gamma(r_D - 1)/r - 1)} \sum_{n=0}^{D-1} \frac{\left(\frac{\gamma(r_D - 1)}{r - 1}\right)^n}{n!} \quad (25)$$

$$P_d = e^{-(2\gamma(r_D - 1)/(2(r - 1) + 3M))} \sum_{n=0}^{D-1} \frac{\left(\frac{2\gamma(r_D - 1)}{2(r - 1) + 3M}\right)^n}{n!}. \quad (26)$$

Note that all the expressions presented assume a perfect minislot synchronization, that is, all the access requests arrive at the base station simultaneously. In fact, this situation is a worst case scenario, as all the access requests suffer the maximum possible interference level. However, this situation keeps the size of the minislots to a minimum and, as they represent an access overhead that is not useful for data transmission, maximizes the data throughput efficiency. It would be possible to fulfill this condition using mobile location techniques [13]. If they are not available, the minislot size is lower bounded by the maximum propagation delay in the system. In a macrocell environment, this value may be significantly greater than the access request size, and the detection probability will be enforced, as not all the received requests will be simultaneous in time. For this case, the expression presented in (26) will represent a lower bound for the detection probability.

Note also that the values presented in Fig. 6 for the detection probability of the receiver filter seem to be low, but they represent a worst-case situation. We are sending a chip sequence in a Rayleigh fading channel using no diversity and only average open loop power control. For example, using antenna diversity of order $D = 2$, and for $P_{fD} = 10^{-3}$ and $r_D = 20$ we have $P_d = 0.93$.

It is proved in [4], [5], and [19] that with only three control minislots, the average number of slots in which n packets

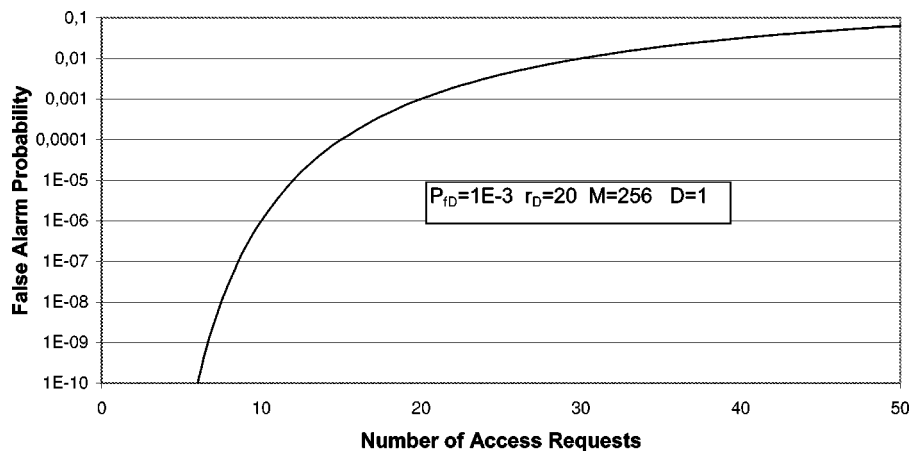


Fig. 5. False alarm probability as a function of the number of access requests.

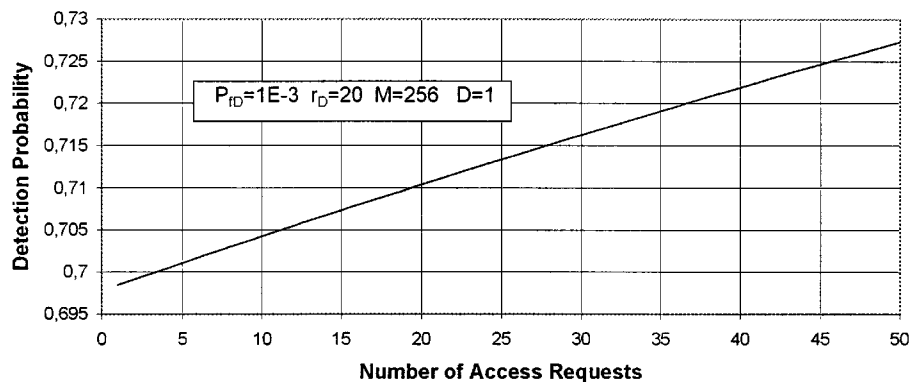


Fig. 6. Detection probability as a function of the number of access requests.

resolve their contention is lower than n , and thus the system throughput is only limited by the data transmission channel rate. Therefore, using only three control minislots, the protocol achieves its best performance, keeping the access overhead loss very small. Henceforth we will always use $m = 3$ for all analytical and simulation purposes. Even more, if messages are long (they consist in more than one transmission packet), it can be seen that with only $m = 2$ minislots, it could be enough to reach the maximum throughput performance [18]. We will also show this feature in Section IV.

D. Probability of Minislot State Misdetection

According to the false alarm and detection probabilities described above, there will be a certain probability of the base station failing to detect the state of each control minislot. We will now describe the expressions of this probability, for all six different error situations possible. We will use E to represent the postdetection empty state, S for the postdetection success state, and C for the postdetection collision state. Fig. 7 shows all these error situations.

First of all, the probability of detecting one or two correlation peaks (the base station detects a successful access), when in fact no user has transmitted its access sequence, is

$$P(S|k=0) = \binom{F}{2} P_f^2 (1-P_f)^{F-2} + F P_f (1-P_f)^{F-1} \quad (27)$$

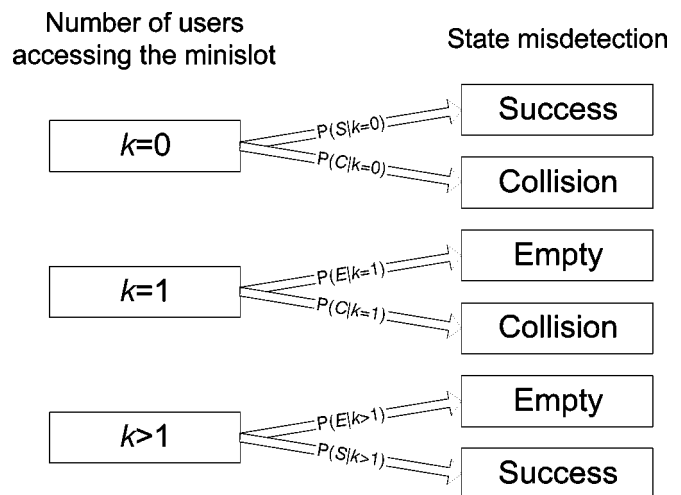


Fig. 7. Possible misdetection situations.

where P_f is the false alarm probability of each detection filter and F is the total number of detection filters. This expression implicitly assumes that the system decides that there has been a single access request transmission when only one correlation peak is detected. This assumption is made supposing that the false alarm probability is much lower than the no-detection probability, which is a reasonable assumption as the false alarm probability is a design parameter.

Second, the probability of detecting a collision (more than two correlation peaks at the detection filter bank output), when no transmission has occurred, is

$$P(C|k=0) = \sum_{k=3}^F \binom{F}{k} P_f^k (1-P_f)^{F-k} = 1 - (1-P_f)^F - P(E|k=0). \quad (28)$$

The probability of detecting no peaks, when a single user has transmitted its sequences, is simply

$$P(E|k=1) = (1-P_d)^2 (1-P_f)^{F-2} \quad (29)$$

where P_d is the detection probability of each matched filter.

In addition, the probability of detecting a collision, when a single user has transmitted, is

$$\begin{aligned} P(C|k=1) &= (1-P_d)^2 \sum_{k=3}^{F-2} \binom{F-2}{k} P_f^k (1-P_f)^{F-k-2} \\ &\quad + 2P_d(1-P_d) \sum_{k=2}^{F-2} \binom{F-2}{k} P_f^k (1-P_f)^{F-k-2} \\ &\quad + P_d^2 \sum_{k=1}^{F-2} \binom{F-2}{k} P_f^k (1-P_f)^{F-k-2}. \end{aligned} \quad (30)$$

This expression has three terms that represent, respectively, the probability that both transmitted sequences from the active user have not been detected, and then three or more false alarms have occurred; the probability that only one of the transmitted sequences has been detected, and then two or more false alarms have occurred; and finally the probability that both sequences have been detected, and one or more false alarms have occurred.

Now, we must evaluate the probability of no peak detection when more than one user has transmitted (i.e., it has been an *a priori* collision). This probability is

$$P(E|k>1) = \sum_{n=2}^N P(E|k=n) P(k=n|k>1) \quad (31)$$

where N is the total number of users in the system.

In this expression, we first know that

$$P(k=n|k>1) = \frac{\lambda^n e^{-\lambda}}{n!} = \frac{\lambda^n}{n!(e^\lambda - \lambda - 1)} \quad (32)$$

where we have assumed that the access requests generated by the users are Poisson-distributed with a total arrival rate λ .

On the other hand, the probability of detecting p peaks at the output of the bank of matched filters, when n users have transmitted their access sequences, is

$$\begin{aligned} P(p|k=n) &= \sum_{f=0}^p \binom{p}{f} P_d^{p-f} P_f^f \sum_{m=0}^{F-p} \binom{p-f+m}{m} (1-P_d)^m \\ &\quad \cdot (1-P_f)^{F-p-m} P_g(p-f+m|n) \end{aligned} \quad (33)$$

where $P_g(a|b)$ is the probability that b users have generated a peaks at the *input* of the bank of matched filters. This probability can be written as

$$P_g(a|b) = \left(\binom{F}{2} \right)^{-1} \binom{F}{a} F_a(b). \quad (34)$$

In this expression, $F_a(b)$ is the number of combinations of b users that are able to generate a peaks. If the number of users is the maximum available for the F filters, this function can be explicitly expressed in a recursive formula (see Appendix I):

$$F_a(b) = \left(\binom{F}{2} \right) - \sum_{n=1}^{a-3} \binom{a}{n} F_{a-n}(b) \quad (35)$$

with the initial values

$$F_2(b) = \begin{cases} 1 & b=1 \\ 0 & b>1 \end{cases} \quad F_3(b) = \begin{cases} 0 & b=1 \\ 3 & b=2 \\ 1 & b=3 \\ 0 & b>3. \end{cases} \quad (36)$$

Finally, by using (33), we can write the probability of detecting p peaks at the output of the bank of matched filters, when n users have transmitted their sequences, as

$$\begin{aligned} P(p|k=n) &= \left(\binom{F}{2} \right)^{-1} \sum_{f=0}^p \binom{p}{f} P_d^{p-f} P_f^f \\ &\quad \cdot \sum_{m=0}^{F-p} \binom{p-f+m}{m} \binom{F}{p-f+m} \\ &\quad \cdot (1-P_d)^m (1-P_f)^{F-p-m} F_{p-f+m}(n). \end{aligned}$$

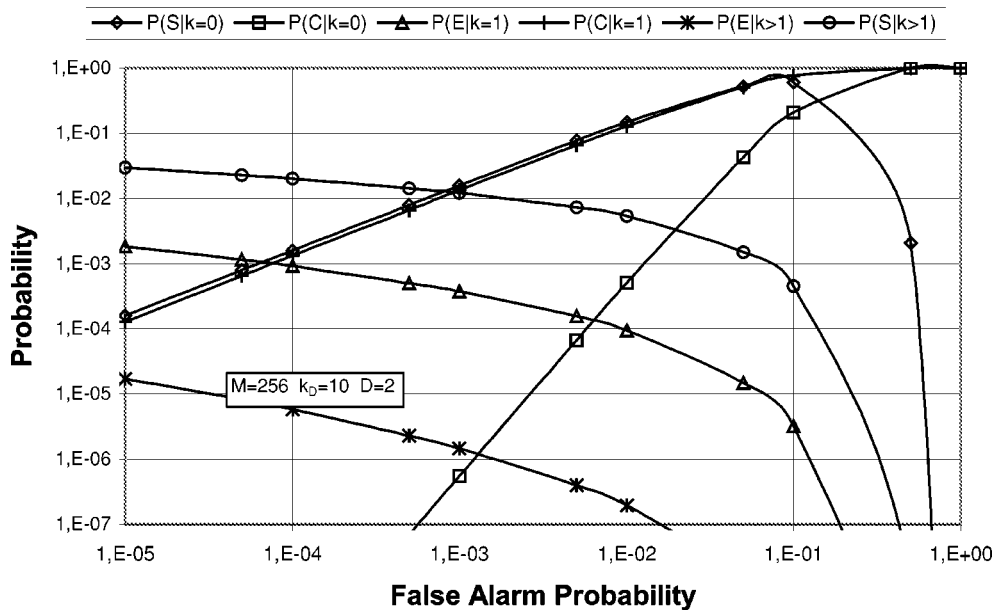
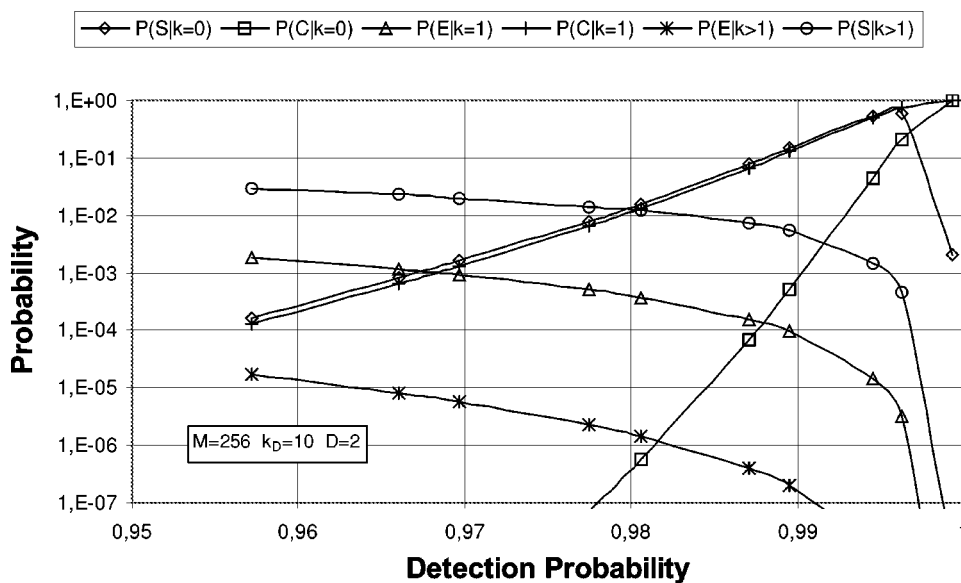
This expression is explained as follows. If we have p peaks at the output, they may be generated by detected real peaks or false alarms. In the expression, f is the number of false alarms generated, and then we have the sum from $f=0$ to $f=p$. $p-f$ is the number of detected peaks. m is then the number of peaks at the *input* of the bank of matched filters that have not been detected. Finally, $F-p-m$ will be the number of filters that must not have a false alarm to get the total p number of output peaks. The other sum in m sweeps all the possible values for this number. The probability for each situation is obtained by multiplying the convenient power of the probabilities of detection and false alarm, and the probability of having $p-f+m$ peaks at the input of the bank of matched filters.

Last, the probability of detecting a success access request when more than one user has transmitted its sequences is

$$P(S|k>1) = \sum_{n=2}^M [P(p=1|k=n) + P(p=2|k=n)] \cdot P(k=n|k>1) \quad (37)$$

which is the probability of detecting one or two peaks at the output when more than one user has transmitted its pair of sequences.

As an example, Figs. 8 and 9 show the minislot state mis-detection probabilities as a function of the receiver filter false alarm and detection probabilities, respectively. The presented

Fig. 8. Minislot state misdetection probabilities as a function of P_f .Fig. 9. Minislot state misdetection probabilities as a function of P_d .

values are for a P_{fD} ranging from 10^{-5} to 1, and with the corresponding P_d values for $M = 256$, $k_D = 10$, $F = 16$, $\lambda = 10$, and antenna diversity of order $D = 2$.

Note that these probabilities are divided into two main groups: those that increase with P_f and P_d , and those that decrease when P_f and P_d increase. It is shown that it is not possible to decrease all the misdetection state probabilities at the same time. This behavior leads to a design decision to choose the value for P_{fD} that makes the protocol perform best. In the following section we will analyze the effect of all the possible error situations on the protocol performance in order to ascertain this value.

E. Protocol Modifications for Error Recovery

Taking into consideration the nonzero probability of the base station failing in the detection of the control minislot states, we

must analyze the necessary protocol adaptations to avoid deadlock situations in a normal data-transmission complete-system steady-state run. The six different error scenarios shown in Fig. 7 are analyzed, and the specific protocol modifications needed are proposed. These modifications will simply consist of corrections and additions to the three sets of rules described in Section II-A, together with changes in the criteria for the base station when sending the broadcast information to the users.

1) *Success Minislot Detection/No User Transmission ($S|k = 0$)*: When this error occurs, all the users, according to rule QDR-1, increment their value for TQ by one extra unit, compared to the one that will be correct. This will represent an “empty” position in the data transmission queue as no user will have its pTQ pointing to this last position. Then, when this empty queue slot reaches one of the last K positions, as no

user will transmit with the corresponding spreading code, the base station will not detect any valid data packet for this code and will detect the error event. At this point, it must set the final-message-bit to 1 for this code. This action will free the queue position and spreading code.

It is also possible that the base station will not detect a valid data packet because of the fading channel propagation or the collision of two or more data packets (see Section II-A-2), so it will be necessary to establish a certain number of consecutive *empty* data slots to consider that this error has occurred. Freeing the data transmission queue position when a user is transmitting in it may cause data collision. Unfortunately, we cannot set this number as high as we would like in order to minimize the probability of this event, because when the error ($S/k = 0$) occurs, this number will be the quantity of wasted slots. It would seem to be a matter of a tradeoff between efficiency and the probability of a data collision occurring.

2) *Collision Detection/No User Transmission* ($C|k = 0$): In this case, all users, according to rule QDR-4, increment their value for RQ by one extra unit, which will also generate an empty position in the collision resolution queue. This event will not cause any critical problem. It will simply mean that no terminal will use one specific spreading code for collision resolution in a certain slot. This fact causes very little loss of contention resolution speed, and in addition does not require any special mechanism to recover from possible deadlock situations.

3) *Empty Minislot Detection/Single User Transmission* ($E|k = 1$): In this situation, only the user that has transmitted its access request is affected. The other users will only do what they are supposed to. The affected one will know that its request has not been detected and will again try to enter the system. Once more, no critical or deadlock situations may appear, only little delay loss is caused, and thus no algorithm modifications are needed.

4) *Collision Detection/Single User Transmission* ($C|k = 1$): According to rule QDR-4, all the users will increase their value for RQ by one extra error unit. The only user that has transmitted its access request will believe that it has collided with another accessing user, and will set its value for pRQ to the last position in the queue. When its turn to resolve the hypothetical collision comes round, it will again send an access request that will certainly not collide again, as no other user has the same position in the collision resolution queue. This situation will cause an extra delay in the message involved in the detection error, but will not cause any deadlock situation nor entail the need for any protocol modification.

5) *Empty Minislot Detection/Multiple User Transmission* ($E|k > 1$): This case is similar to the one presented above under subsection 3, but with more than one user involved. All the users that have transmitted an access request know that their sequences have not been detected, and then they will again try to enter the system. The rest of the users are not affected as they do not change any of their counter values. Again, we have no efficiency loss other than the extra delay suffered by the messages of the affected users, and no deadlock error is possible.

6) *Success Minislot Detection/Multiple User Transmission* ($S|k > 1$): This is the most critical of the possible error sit-

uations. If more than one user transmits an access request and they receive success state information from the base station, all of them will believe that their own request has succeeded. They will thus set their pTQ pointers to the same position in the data transmission queue. When they start the data transmission, all the packets will collide and the data transfer will become impossible.

We present a possible mechanism to prevent such users from getting into an endless collision situation. This mechanism will preserve the performance of the users that have not been affected by the state detection error, so they will maintain the same transmission characteristics. Only the affected users will suffer from extra delay in their messages.

The idea is as follows: any time a user detects a certain number of consecutive erroneous data transmissions, it will enter a special *backlogged* state. In this state, instead of transmitting its data packets with probability one, it will do so with a certain probability P_b . This will make it possible to share the same spreading code within the group of colliding users. According to this probability, when the base station detects a certain number of consecutive empty slots after receiving two or more packets with the final-message-bit set to one, it will assume that all the users involved have finished their transmission and free the data transmission queue position (i.e., the spreading code). There will be a nonzero probability of the base station freeing the spreading code before all the users have finished their message transmission, but we can make this probability as small as we like by increasing the number of empty slots the base station has to encounter before freeing the code. This number is in fact the same as defined in subsection 1 above, and must be greater than the number of slots to enter the *backlogged* state. Even in the event of this situation occurring, the terminal that has its message pending transmission will have to reset its connection after a certain number of consecutive erroneous data transmissions.

Furthermore, probability P_b can be dynamically adjusted to maximize the throughput of the *backlogged* users as in a normal adaptive slotted-ALOHA procedure. The mechanism must reduce P_b when errors are frequent, and increase P_b when data packets are received correctly.

IV. SIMULATION RESULTS AND COMPARISONS

Bearing all the ideas presented in mind, computer simulations were carried out to validate the protocol operation. First of all, to validate the analytical model presented in Section III-A, we assumed a perfect minislot state detection. Fig. 10 shows the comparison between the analytical evaluation and the simulation results of the total delay for a system with $N = 100$ data terminals, average message length of 6400 bits, CDMA channel with $K = 16$ spreading codes (simultaneous transmissions), and a spreading factor $S_f = 64$. The slots are of $L = 640$ bits and have $m = 3$ access minislots. We have assumed a perfect closed-loop power control, a single-cell operation environment, and used the Gaussian hypothesis for the interference power evaluation in the calculation of the BLER.

The choice for the number of maximum simultaneous transmissions allowed in the system is not arbitrary. Indeed, with the

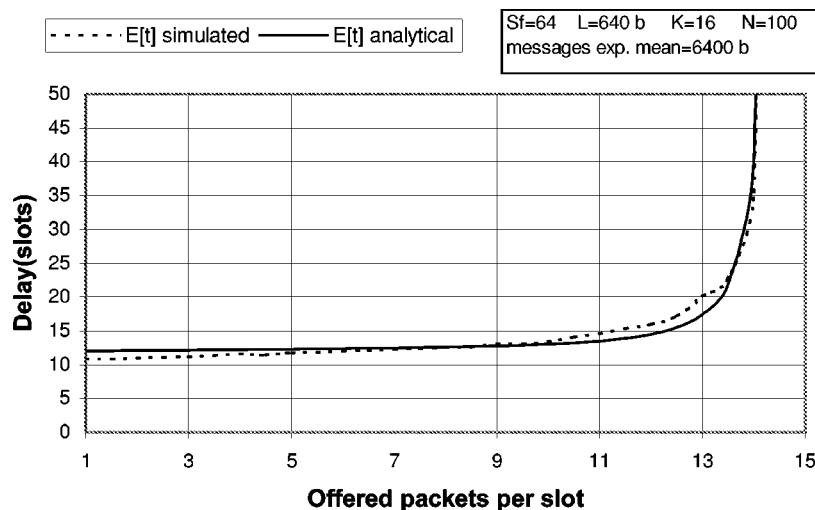


Fig. 10. Analytical and simulation delay results for the DQRAP/CDMA.

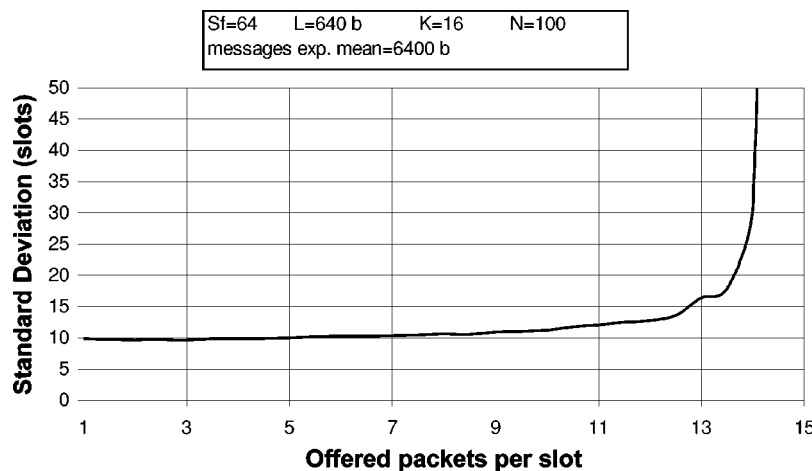


Fig. 11. Standard deviation of the message delay in DQRAP/CDMA.

above channel assumptions, for a number k of simultaneous transmissions, the average number of bits effectively sent through the channel, \bar{L} , is

$$\begin{aligned} \bar{L} &= kL(1 - \text{BLER}) \\ &= kL \left(1 - \left(1 - \left(1 - \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{3S_f}{2(k-1)}} \right) \right) \right)^L \right) \\ &= kL \left(1 - \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{3S_f}{2(k-1)}} \right) \right)^L. \end{aligned} \quad (38)$$

This number gives the average number of bits per slot transmitted without errors, that is, bits that will not need to be retransmitted. For a given set of channel parameters, we can numerically find the optimum number for k that maximizes \bar{L} . In particular, for the proposed channel parameters, $k = 16$ is the optimal choice and the maximum achievable throughput is about $14.3L$ bits/slot. This value is therefore used for obtaining the results shown in Fig. 10.

The results shown in Fig. 10 fit the analytical model correctly. Fig. 11 shows the standard deviation of the message delay, obtained by simulations, while Fig. 12 shows the throughput of the system versus the offered traffic load, also from simulations.

These figures show two very important features of the protocol. 1) The maximum stable throughput achieved by the system is near the optimum (about $13.5L$ bits/slot for a channel capacity of $14.3L$ bits/slot), also maintaining a small mean delay for offered traffic up to this maximum value. 2) The standard deviation of the delay is bounded by its mean value, which implies that the distribution of the particular delays of each message has a limited range.

Subsequently, we supposed that all the access requests suffer a Rayleigh fading propagation (the terminals are only able to use an open loop power control) and applied all the protocol modifications proposed in Section III-E. Fig. 13 shows the delay comparison between the ideal and the nonideal situation with $P_{fD} = 10^{-3}$ and $k_D = 20$ and using antenna diversity of order $D = 2$. We can notice the small loss caused by the misdetection in the control minislots and thus the robustness of the protocol.

Let us note that for heavy traffic conditions, for example, 13 offered packets per slot, the average number of simultaneous access requests is about 2, and this represents a $P_f \cong 10^{-24} \cong 0$ and a $P_d = 0.93$.

Next, we carried out a comparison to other medium access protocols designed for a CDMA environment. Slotted-ALOHA/CDMA [9] has been extensively used as

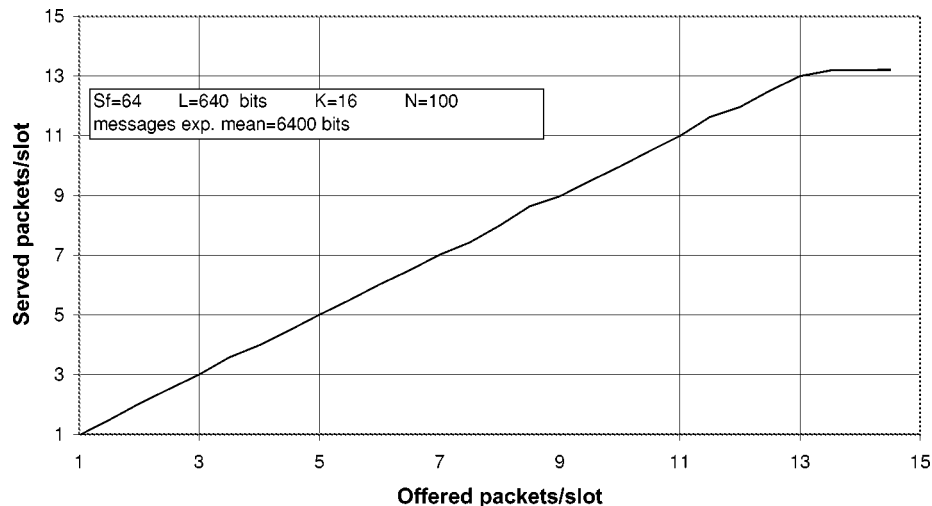


Fig. 12. Throughput of DQRAP/CDMA.

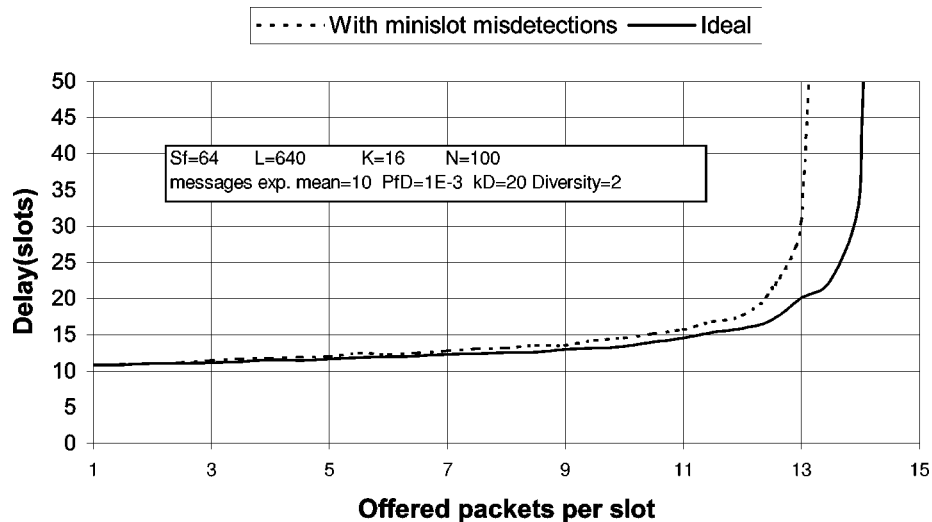


Fig. 13. DQRAP/CDMA delay. Ideal versus possible minislot state misdetection.

an access scheme. With this scheme, whenever a terminal has data to send, it makes the transmission immediately. A more sophisticated access scheme proposed to date is ISMA (inhibit sense multiple access) [14]. In this protocol, the base station informs the users about the occupied spreading codes using a feedback information channel. This avoids data collisions except for the first access packets of each user.

In order to have a performance reference of DQRAP/CDMA with respect to these other well known protocols, Fig. 14 shows the delay comparison between a slotted-ALOHA/CDMA system with $N = 100$ users that have an assigned spreading code, an ISMA system with the same number of users but using only $K = 32$ simultaneous spreading codes, and a system using DQRAP/CDMA protocol using only $K = 16$ available spreading codes. Channel and traffic conditions are the same for all three systems (spreading factor $S_f = 64$, slots of length $L = 640$ bits, and Poisson-generated messages of exponentially distributed length with mean 6400 bits). We can see that DQRAP/CDMA using only 16 spreading codes (receivers) for all the users outperforms the other protocols using a greater number of codes, in terms of the maximum

stable throughput maintaining good delay characteristics. It can manage heavier traffic load without entering an instability region and keeping the same low delay for light traffic load.

Fig. 15 illustrates the fact that, when messages are long in terms of number of data packets needed to send per message, the number of total access requests decreases remarkably, and it should be possible to use only $m = 2$ control minislots and achieve a very close performance result. This feature is shown by the little performance difference between $m = 3$ and $m = 2$ cases when messages have an average length of 10 packets.

In order to show the inherent capacity of DQRAP/CDMA to manage mixed traffic sources, we considered a mixed voice-data traffic scenario. The scenario consists of a variable number of voice ON-OFF stations with an activity factor of 0.45, generating bursts of 16 kbits/s with a maximum allowed packet delay of 20 ms and a maximum packet loss ratio of 1%, together with data stations (using a representative IP traffic model) generating packets with Poisson arrival rate $\lambda = 0.05$ packets/s and messages sizes defined by the *pdf* shown in Fig. 16. The number of maximum simultaneous transmissions has been maintained in 16 (the optimum one when only data packets of 640 bits are

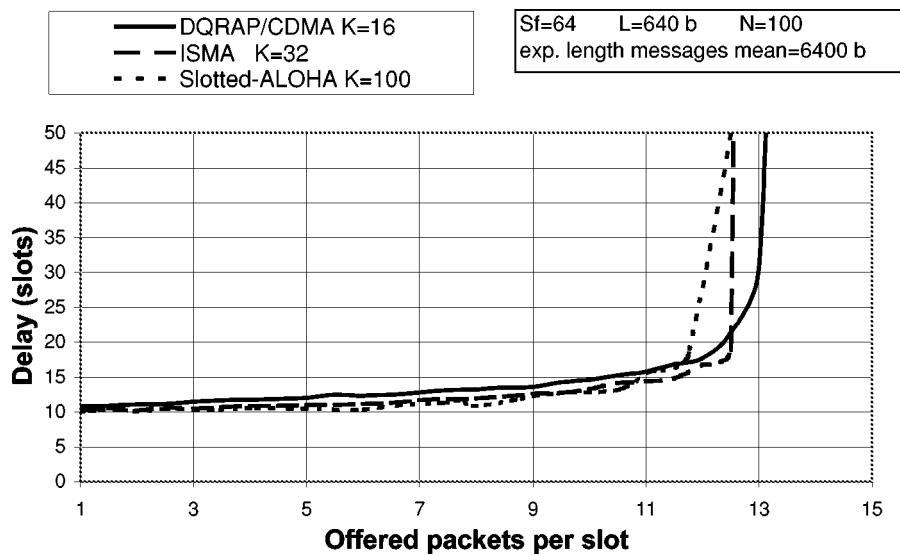


Fig. 14. Delay comparison between protocols.

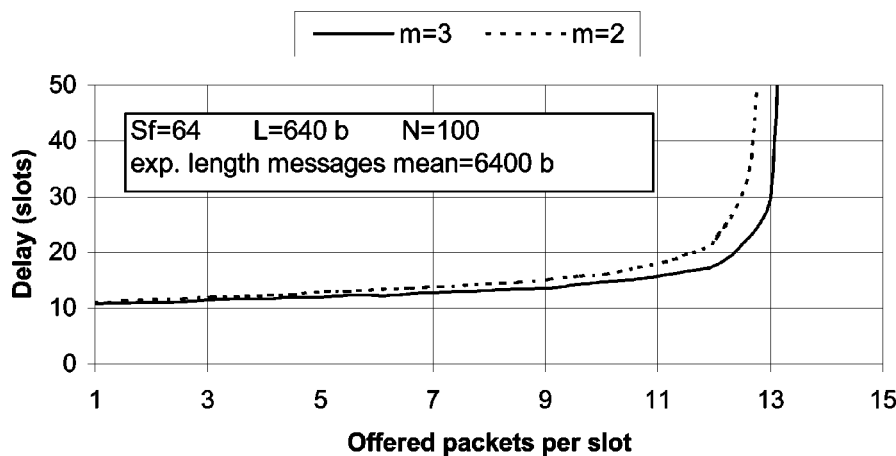


Fig. 15. Performance variation with the number of minislots.

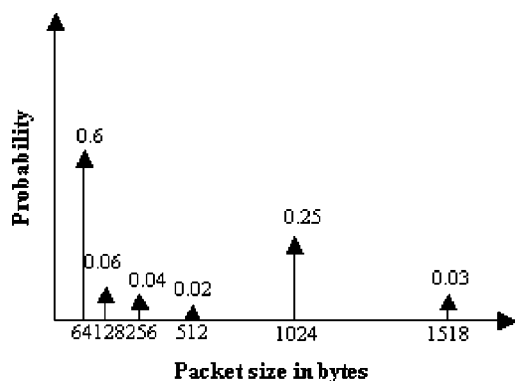


Fig. 16. IP data packets pdf.

transmitted). Fig. 17 shows the message delay performance for the described scenario. The curves in this figure are drawn for data traffic loads that maintain the voice packet loss ratio up to the maximum allowed. It can be noticed that the behavior of the data packet delay performance is in practice the same for different combinations of heterogeneous traffic sources. Actually, the near-flat operative region of each curve simply decreases as

far as the number of voice users increases. This fact outlines the inherent adaptation of the protocol, without modifications, to a mixed traffic source situation.

At this point, we will summarize the main advantages of DQRAP/CDMA with respect to other MAC protocols designed for CDMA environments. DQRAP/CDMA avoids the typical instability problems of ALOHA-based protocols, putting all the exceeding load in queues and maintaining the optimum interference level in the system. Due to the lower number of spreading codes required to achieve a given performance, the number of simultaneous transmissions is kept under control and upper bounded, and therefore DQRAP/CDMA performs better than other ALOHA-based protocols in terms of both a lower number of retransmissions (saving energy) and a lower interference generation in cellular CDMA-based systems. On the other hand, as pointed before, it is based on distributed queues, and then it is inherently adapted for supporting mixed-type traffic sources. Moreover, in that respect, the application of appropriate scheduling policies to the queues of the protocol should allow the system to support services with a certain guaranteed QoS, which should be operative in the future third-generation mobile communication systems.

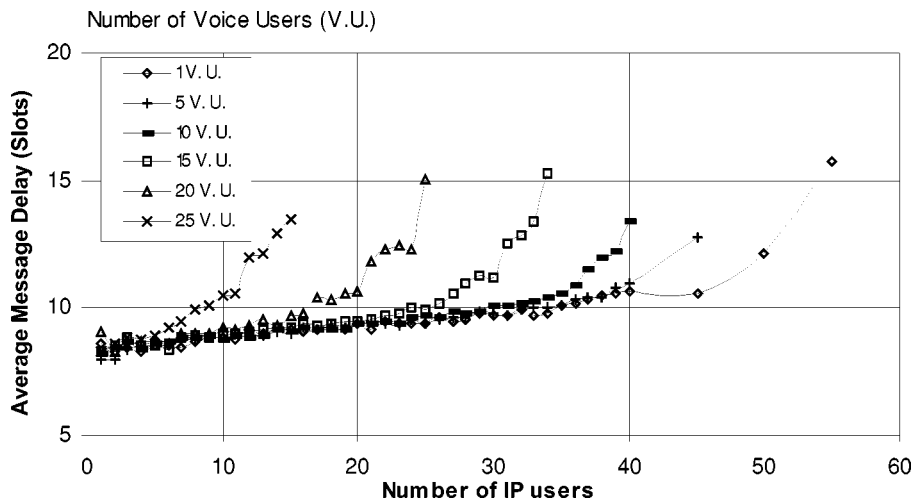


Fig. 17. IP message delay with mixed IP-voice traffic sources.

Finally, we may note that even the conceptual complexity of DQRAP/CDMA seems to be rather considerable, the protocol algorithm is quite simple to implement, and the computational load added is minimum (only integer simple operations are required). Moreover, it would simplify, to some extent, the base station complexity, as it reduces the number of total receivers needed to manage a certain number of mobile terminals.

V. CONCLUSION

A proposal for a near-optimum random access protocol for a CDMA environment suitable for the future third generation mobile communication systems has been presented. An analytical model has been introduced, and the results obtained match the ones obtained by computer simulations well. It has been shown that the protocol has good delay and stability characteristics, maintaining the standard deviation of the message's delay bounded by its mean value and achieving a nearly optimum maximum stable throughput, for given channel characteristics. It is therefore a suitable proposal for improving the use of the capacities of random access channels in a reverse link.

A receiver scheme for the detection of access requests has been proposed and analyzed, and the misdetection state probabilities have been derived. The protocol's sensitivity to errors in the detection of the state of the control minislots has been studied. Protocol modifications have been introduced to manage the possible error scenarios, showing great robustness and little efficiency loss in realistic channel conditions.

It has been shown that the protocol outperforms other widely used multiple access schemes in terms of the maximum stable throughput and the delay characteristics.

APPENDIX I

We are to find the expression of the number of combinations of b users that are able to generate a peaks if they are assigned a unique pair of sequences between F different available ones. We must make an abstraction of the problem as follows.

Let $1, 2, 3, \dots, a$ be the first a integer numbers. That is, we number the received peaks from 1 to a .

Let $a(a - 1)/2$ be the different possible pairs we can create. Each pair represents one user. Let this be all the possible groups of pairs of numbers, that is, all the possible groups of users having 1 to $a(a - 1)/2$ users per group. We must evaluate, for any b from 1 to $a(a - 1)/2$, which of these groups contain at least once all the numbers from 1 to a , being able to repeat the numbers as many times as desired, and how many users there are in each group.

Calling this number $F_a(b)$, we will suppose that we know the value of $F_{a-c}(k)$ for any value $c > 0$, and with these values we will evaluate the function for $c = 0$ (the target value).

We calculate all the different groups of b users we can make from the total $a(a - 1)/2$ possible users, and then we subtract those that do not have all the numbers from 1 to a . Which groups do not fulfill this condition? First of all, those that leave one number unselected, which will be the number of pairs that generate $a - 1$ peaks multiplied by the a positions where we can locate the blank. Then, we must subtract the pairs that leave two unselected numbers multiplied by the number of combinations leaving two blanks of a number, and so on. The result is thus

$$\begin{aligned}
 F_a(b) &= \binom{a}{2} - \left[\binom{a}{1} F_{a-1}(b) + \binom{a}{2} F_{a-2}(b) \right. \\
 &\quad \left. + \dots + \binom{a}{a-3} F_3(b) \right] \\
 &= \binom{a}{2} - \sum_{n=1}^{a-3} \binom{a}{n} F_{a-n}(b).
 \end{aligned}$$

We have explicitly eliminated the terms for $b < 2$ and $a < 4$ because they are zero. We only need the initial values to evaluate the recursive expression. These are

$$F_2(b) = \begin{cases} 1 & b = 1 \\ 0 & b > 1 \end{cases} \quad F_3(b) = \begin{cases} 0 & b = 1 \\ 3 & b = 2 \\ 1 & b = 3 \\ 0 & b > 3. \end{cases}$$

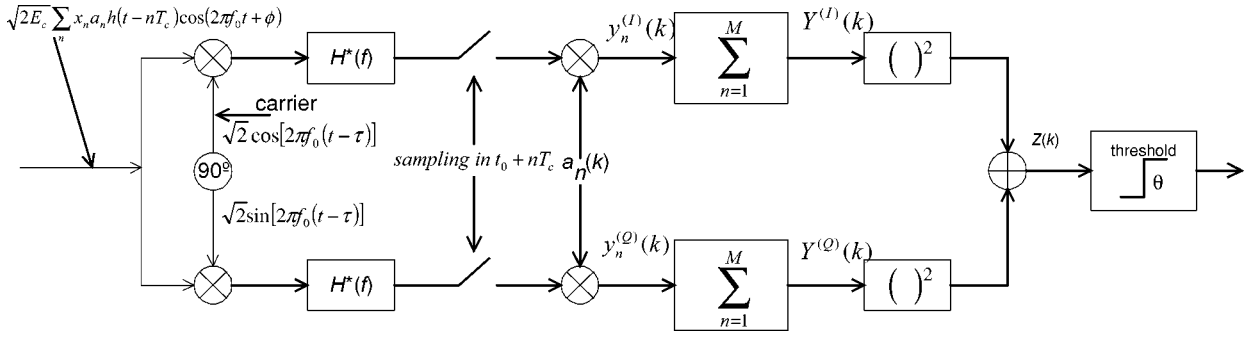


Fig. 18. Receiver structure for each receiver filter.

APPENDIX II

It is shown in [16] that the optimum receiver scheme for a sequence detection filter is the one shown in Fig. 18. Each branch of the receiver presented in Fig. 3 has this structure.

As shown in the figure, for a given user k , the input signal follows the expression $s(t) = \sqrt{2E_c(k)} \sum_n x_n(k) a_n(k) h(t - nT_c) \cos(2\pi f_0 t + \phi_k)$, where E_c represents the energy per chip, $h(t)$ is the impulse response of the channel, and x_n is the information that modulates the code sequence a_n . We chose $x_n(k) = 1$ for all n , that is, we send a single bit, without modulating $a_n(k)$. With these assumptions, the expected value of the correlation between the input signal and the local copy of the sequence, for the in-phase and quadrature sequences $Y(k)$, has the following expression [16]:

$$\begin{aligned} E[Y^{(I)}(k)] &= M\sqrt{E_c(k)}(\cos \phi_k)R(\tau) \\ E[Y^{(Q)}(k)] &= M\sqrt{E_c(k)}(\sin \phi_k)R(\tau) \end{aligned} \quad (39)$$

where

$$R(\tau) = \int_{-\infty}^{\infty} |H(f)|^2 \cos(2\pi f\tau) df \quad (40)$$

and M is the number of chips in the sequence (if they correspond to one bit, this value will be equal to the spreading factor). The matched filter output peak corresponds to $\tau = 0$, which implies that $R(\tau) = 1$ for any input filter $|H(f)|$. As both Y signals are squared and added, the phase term ϕ_k becomes irrelevant, always supposing that this value remains constant during the M -chip transmission time.

On the other hand, the variance of both components is [16]

$$\text{Var}(Y^{(I)}) = \text{Var}(Y^{(Q)}) = \frac{MI_0}{2} \quad (41)$$

where

$$I_0 = N_0 + \sum_{j \neq k} E_c(j) \int \frac{|H(f)|^4}{T_c} df. \quad (42)$$

I_0 represents the thermal noise N_0 , plus the total interference caused by the rest of users. For a time-limited filter, the value of the integral in (42) is $2/3$. Thus, assuming a power control that maintains the same E_c for all users, the variance of both in-phase and quadrature components is

$$\text{Var}(Y) = \frac{MI_0}{2} = \frac{M}{2} \left(N_0 + \frac{2}{3}(k-1)E_c \right) \quad (43)$$

with k being the number of simultaneous access request transmitted. To find the detection and false alarm probabilities of the receiver scheme, we must take into account the propagation

channel conditions. We will consider a Rayleigh fading environment.

Note that the decision variable is $Z = [Y^{(I)}]^2 + [Y^{(Q)}]^2$. It can be proved that the diagram shown in Fig. 18 is optimal for signals with unknown phase, according to either the Bayes or the Newman–Pearson optimality criteria [17]. Using the latter, the system design consists in fixing the decision threshold value θ to obtain a certain allowable false alarm probability. The criterion guarantees that the chosen value is that which maximizes the detection probability for that false alarm probability value. These probabilities are obtained from integrating two likelihood functions of Z , depending on the initial possible hypothesis: $p_0(Z)$ under the assumption that no signal has been transmitted, and $p_1(Z)$ under the assumption that the target sequence has been transmitted. Using antenna and postdetection diversity of order D , that is (similarly to the time postdetection integration used in [16]), adding the contributions of D independent Z signals coming from the same number of different antennas and receivers, these functions are given by

$$p_0(Z) = \frac{Z^{D-1} e^{-Z/V}}{(D-1)!V^D} \quad (44)$$

$$p_1(Z) = \frac{Z^{D-1} e^{-(Z/V + S^2\sigma^2)}}{(D-1)!(V + S^2\sigma^2)^D} \quad (45)$$

where V is twice the variance of each component Y , and S^2 is the mean square, which is obtained as the sum of the squares of the in-phase and quadrature component means. Defining $V_F = M(I_0 + ME_c)$, the likelihood functions are finally

$$p_0(Z) = \frac{Z^{D-1} e^{-Z/V}}{(D-1)!V^D} \quad p_1(Z) = \frac{Z^{D-1} e^{-Z/V_F}}{(D-1)!V_F^D}. \quad (46)$$

The false alarm and detection probabilities are obtained by evaluating the integral of the corresponding function from the threshold value θ to infinity, thus giving

$$P_f = \int_{\theta}^{\infty} p_0(Z) dZ = e^{-\theta/V} \sum_{k=0}^{D-1} \frac{(\theta/V)^k}{k!} \quad (47)$$

$$P_d = \int_{\theta}^{\infty} p_1(Z) dZ = e^{-\theta/V_F} \sum_{k=0}^{D-1} \frac{(\theta/V_F)^k}{k!}. \quad (48)$$

REFERENCES

- [1] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall International, 1992.
- [2] B. S. Tsybakov and N. B. Likhanov, "Upper bound on the capacity of a random multiple access system," *Problems Inform. Transmission*, vol. 23, no. 3, pp. 224–236, 1987.
- [3] T. Towsley and P. O. Vales, "Announced arrival random access protocols," *IEEE Trans. Commun.*, vol. COM-35, pp. 513–521, May 1987.

- [4] W. Xu and G. Campbell, "A near perfect stable random access protocol for a broadcast channel," in *IEEE Proc. ICC'92*, vol. 1, pp. 370–374.
- [5] —, "DQRAP—A distributed queueing random access protocol for a broadcast channel," presented at the SIGCOMM'93, San Francisco, Sept. 14, 1993.
- [6] E. Dahlman, P. Beming, J. Knutsson, F. Ovesjö, M. Persson, and C. Roobol, "WCDMA—The radio interface for future mobile multimedia communications," *IEEE Trans. Veh. Technol.*, vol. 47, Nov. 1998.
- [7] "UMTS terrestrial radio access: Concept evaluation (UMTS 30.06)," ETSI Tech. Rep. 101 146, version 3.0.0, Dec. 1997.
- [8] "Requirements for the radio interface(s) for future public land mobile telecommunication systems (FPLMTS)," Recommendation ITU-R M.1034, 1994.
- [9] A. Chockalingam, W. Xu, and L. Milstein, "Performance of a multi-channel packet CDMA protocol in a fading environment," in *Conf. Rec., IEEE Veh. Technol. Conf., VTC'97*, 1997.
- [10] L. Kleinrock, *Queueing Systems*. New York: Wiley, 1976.
- [11] X. Zhang and G. Campbell. (1993, Aug.) Performance analysis of distributed queueing random access protocol—DQRAP. DQRAP Research Group Rep. 93-1, Computer Sci. Dep., Illinois Inst. Technol. [Online]. Available: <http://www.iit.edu/~dqrapp/html/papers2.html>
- [12] M. B. Pursley, "Performance evaluation for phase-coded spread-spectrum multiple-access communication—Part I: System analysis," *IEEE Trans. Commun.*, vol. COM-25, pp. 795–799, Aug. 1977.
- [13] T. S. Rappaport, J. H. Reed, and B. D. Woerner, "Position location using wireless communications on highways of the future," *IEEE Commun. Mag.*, pp. 31–44, Oct. 1996.
- [14] J. Pérez, R. Agustí, and O. Sallent, "Performance analysis of an ISMA CDMA packet data network," in *Proc. IEEE Veh. Technol. Conf., VTC'99 Fall*, Amsterdam, Sept. 1999.
- [15] M. J. Karol, Z. Liu, and K. Y. Eng, "Distributed-queueing request update multiple access (DQRUMA) for wireless packet (ATM) networks," in *Proc. ICC'95*, Seattle, WA, pp. 1224–1231.
- [16] A. J. Viterbi, *CDMA Principles of Spread Spectrum Communication*. Reading, MA: Addison-Wesley, 1995.
- [17] J. Neyman and E. S. Pearson, "On the problem of the most efficient tests of statistical hypotheses," 1933.
- [18] C.-T. Wu and G. Campbell, "Extended DQRAP (XDQRAP), a cable TV protocol functioning as a distributed switch," DQRAP Research Group Rep. 94-2.
- [19] W. Xu and G. Campbell, "DQRAP—A distributed queueing random access protocol for a broadcast channel," *Computer Commun. Rev.*, vol. 23, no. 4, pp. 270–278, Oct. 1993.

Luis Alonso (M'99) received the Engineer degree in telecommunications from the Universitat Politècnica de Catalunya (UPC), Spain, in 1997.

He joined the Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona, Spain, as Visitant Professor in 1998. In 1999, he joined the Escola Universitària Politècnica del Baix Llobregat, Spain, where he became Assistant Professor. He is currently doing his Ph.D. thesis about medium access protocols, scheduling algorithms, packet radio techniques, and spread-spectrum systems for mobile communications.

Ramon Agustí (M'78) was born in Riba-roja d'Ebre, Spain, on August 15, 1951. He received the Engineer of Telecommunications degree from the Universidad Politècnica de Madrid, Spain, in 1973, and the Ph.D. degree from the Universitat Politècnica de Catalunya, Spain, 1978.

In 1973, he joined the Escola Tècnica Superior d'Enginyers de Telecomunicació de Barcelona, Spain, where he became Full Professor in 1987. He has been working in the field of digital communications with particular emphasis on digital radio, both fixed radio relay, and mobile communications. He has also been concerned with the performance analysis and development of frequency-hopped spread-spectrum systems. He participated in the COST 231, RACE, and ACTS European research programs, and currently is participating in the IST program. His research interests are in the area of mobile communications with special emphasis on CDMA systems and packet radio networks.

Oriol Sallent (M'98) received the Engineer and Doctor Engineer degrees in telecommunication from the Universitat Politècnica de Catalunya (UPC), Spain, in 1994 and 1997 respectively. He received the Doctorate Award from the Telecommunication Engineer Association of Spain in 1997 for his Ph.D. dissertation on multiple access protocols for CDMA-based systems.

He joined the Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona, where he became Assistant Professor in 1994 and Associate Professor in 1998. His research interests are in the field of mobile communication systems, especially packet radio techniques and spread-spectrum systems.