A Novel MAC Protocol Based on Distributed Queues for CDMA Communication Systems

Luis G. Alonso, Ramón Agustí, Oriol Sallent
Department of Signal Theory and Communications - Technical University of Catalonia (UPC)
c) Jordi Girona 1-3. Campus Nord. Edifici D4. 08034.-Barcelona. Spain
Tel.: 34-93-401 71 95 Fax: 34-93-401 72 00 e-mail: [luisg, ramon, oriol]@tsc.upc.es

Abstract[*] **–** This paper presents and analyses a new near–optimum medium access control (MAC) protocol. The proposed access scheme is suitable for a CDMA mobile communication environment and minimises the total number of spreading codes needed to achieve a certain throughput. It also has a delay performance approaching that of an ideal optimum M/M/K system, where K is the number of spreading codes being used. 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 analysed 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 **–** Multi-access communication, Protocols, Code division multi-access, Mobile communications.

1 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) [12]. 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 centralised strategy is that it allows the designer to totally control the behaviour of the system. It is possible to give priority to terminals with strict quality requirements, such as tight delay bounds, instead of simply maximising the overall throughput. However, high complexity algorithms, a great amount of signalling 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], 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 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 one of the most likely candidates for third generation mobile telecommunication systems. Schemes based on Wide-band CDMA (WCDMA) [5] have been chosen as radio interfaces in the standardisation body in Japan (ARIB), and also in Europe by the ETSI for the

---

[*] This research was supported by CYCIT project TIC 98-684

UMTS Terrestrial Radio Access (UTRA) [6]. This access scheme is also being considered in the International Mobile Telecommunication 2000 (IMT-2000) [7] 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 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 modelled 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 and thus completely eliminating collisions. This behaviour is the key to its good delay and throughput performance. The main advantage of this protocol in a CDMA environment is that, for given channel characteristics, it minimises the number of spreading codes needed to reach a certain throughput. Therefore, it may allow use of a very good set of spreading sequences (orthogonal instead of pseudo-random or others with good correlation properties), thus reducing the total interference level and allowing an increase in system capacity.

The near-optimum characteristic of this protocol can be seen as follows: given certain CDMA channel characteristics (i.e., spreading factor, bits per slot, fading and interference model, diversity, coding, ARQ strategy, etc.) there is always an optimum number of simultaneous transmissions that gives the maximum effective throughput. DQRAP/CDMA allows this number of simultaneous transmissions to be kept in the system, putting all the data that exceed this maximum in a transmission queue.

In order to assess the DQRAP/CDMA scheme under realistic conditions, a receiver scheme for the control minislot detection was proposed and analysed. Expressions for the minislot 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 with other MAC schemes extensively studied in the open literature such as Slotted-ALOHA/CDMA [8] and ISMA/CDMA [11]. The results obtained show a significant improvement in the system delay and throughput performance.

The paper is organised as follows. The protocol description is detailed in Section 2. In Section 3, the analytical model is presented and studied. Expressions for the total system delay are also derived in this section. Section 4 explains and analyses the proposed scheme for the control minislot state detection. In this section, protocol algorithm modifications are also introduced to recover from errors in the minislot detection. Section 5 shows computer simulation results and comparisons with other protocols. Finally, Section 6 is devoted to the conclusions.

2 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 minislot synchronisation. 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 minislot 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.

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).

2.1 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.

2.1.1 QDR (Queueing Discipline Rules)

1. 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.
2. 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.
3. If RQ>0, each station reduces the value of RQ by $min(\mathrm{RQ}, K)$ units.
4. 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.
5. 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.

2.1.2 DTR (Data Transmission Rules)

1. If $\mathrm{TQ}<K$, 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 $K_{\mathrm{TQ}+1}$. 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.
2. If a station has pTQ>0 and $\mathrm{pTQ}\leq K$, 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.

2.1.3 RTR (Request Transmission Rules)

1. If $\mathrm{RQ}<K$, 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 $K_{\mathrm{RQ}+1}$ and transmits a request in it.
2. If a station has pRQ>0 and pRQ≤K, the station randomly selects one of the control minislots of the spreading code K_{pRQ} and transmits a request in it.

2.2 Example

The example shown in Figure 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.

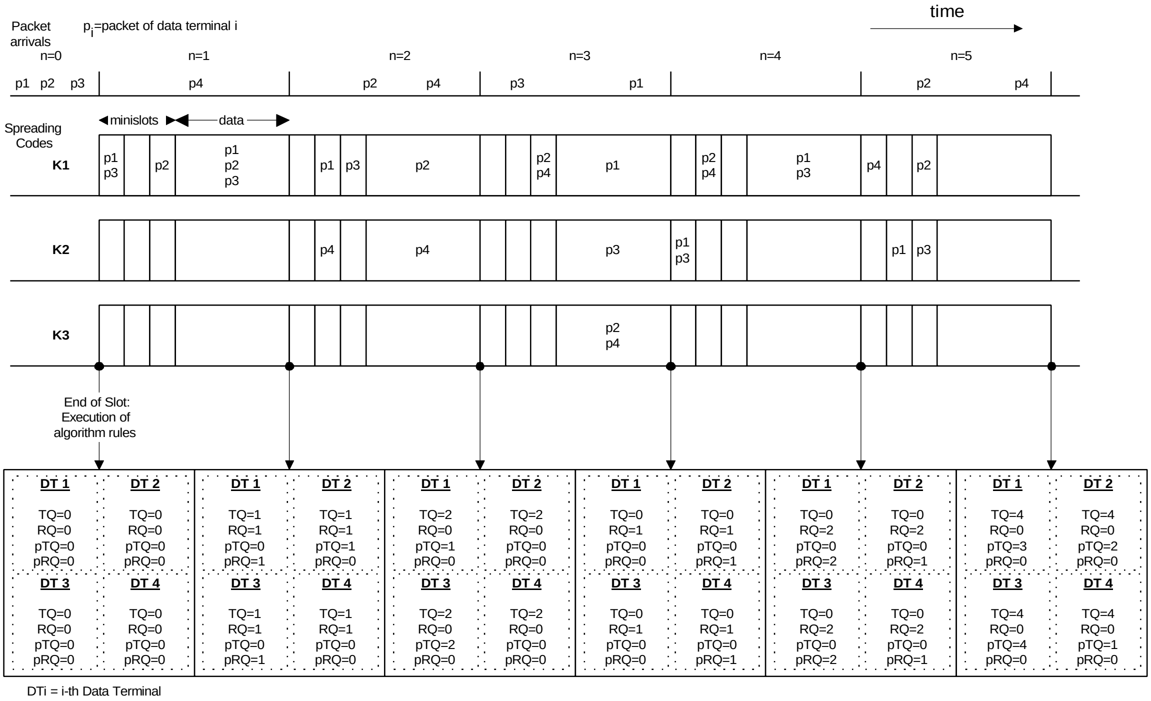

Figure 1. Example of DQRAP/CDMA protocol operation

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 *p3* 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.

3 PROTOCOL MODEL AND ANALYSIS

The DQRAP/CDMA protocol can be modelled as shown in Figure 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. Normalising the time axis in slot units, this 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*).

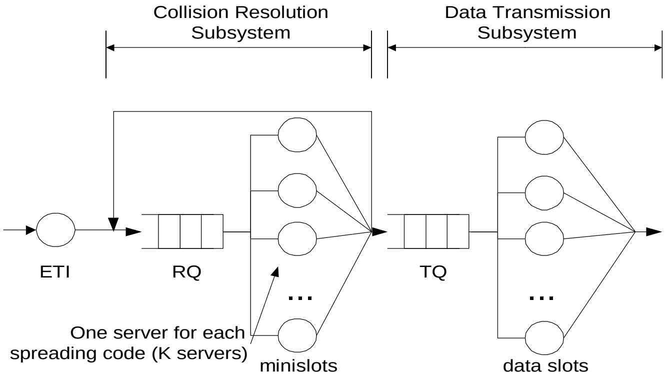

Figure 2. Model of DQRAP/CDMA protocol

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.

3.1 Delay analysis

Using the classical queue theory [9], it can be proved [14] that the total delay for a message (t_T) can be expressed as:

$$E[t_T] = \frac{1}{2} + \left[\frac{1}{\ln\left(\frac{1}{1-e^{-l/m}}\right)} + \frac{P_{KRQ}}{K\ln\left(\frac{1}{1-e^{-l/m}}\right)(1-r_{RQ})} \right] + \left[\frac{1}{m(1-BLER)} + \frac{P_{KTQ}}{Km(1-BLER)(1-r_{TQ})} \right] +$$

$$+ \frac{\sum_{n=0}^{K-1} \frac{(Kr_{TQ})^n}{n!}}{\sum_{n=0}^{K-1} \frac{(Kr_{TQ})^n}{n!} + \frac{(Kr_{TQ})^K}{K!(1-r_{TQ})}} \left(1-e^{-l}(1+l)\right) \qquad (1)$$

where *BLER* is the average block error probability, that can be expressed as [10]:

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

S_f is the spreading factor, k is the number of simultaneous data transmissions, L is the number of bits contained in the packets sent during a time slot, l is the message arriving rate (messages per slot), m the number of minislots per spreading code and per slot, $1/m$ is the average number of packets per message, P_{KTQ} and P_{KRQ} are the probabilities of delay (when a message arrives to the queue system, all K servers are occupied) for the transmission and request subsystems respectively, according to an Erlang C distribution, and finally:

$$r_{TQ} = \frac{l}{Km} \qquad (3)$$

$$r_{RQ} = \frac{l}{Km_{RQ}} \qquad (4)$$

where

$$\frac{1}{m_{RQ}} = \left[\ln\left(\frac{1}{1-P(l)}\right)\right]^{-1} \qquad (5)$$

3.2 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 synchronisation, equalisation 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: (i) empty, that is, no energy is received; (ii) success, that is, a single burst from any terminal has been detected; and (iii) collision, 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 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.

Figure 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.

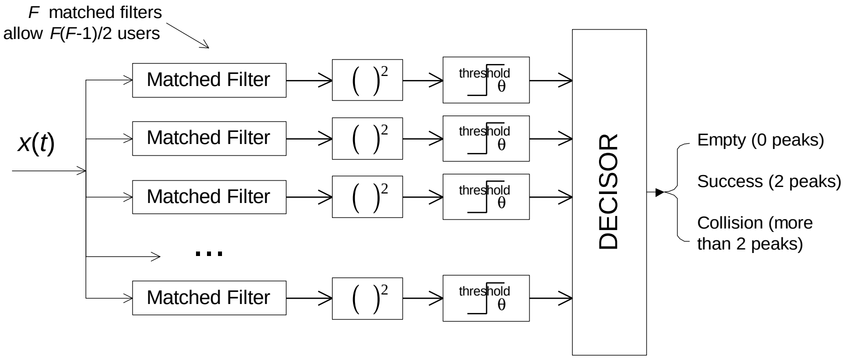

Figure 3. Structure of the minislot receiver at the base station

3.3 Probability of minislot state misdetection

According to the false alarm and detection probabilities of the matched filters described above, there will be a certain probability of the base station failing to detect the state of each control minislot. We will use *E* to represent the post-detection empty state, *S* for the post-detection success state and *C* for the post-detection collision state. Figure 4 shows all the possible error situations.

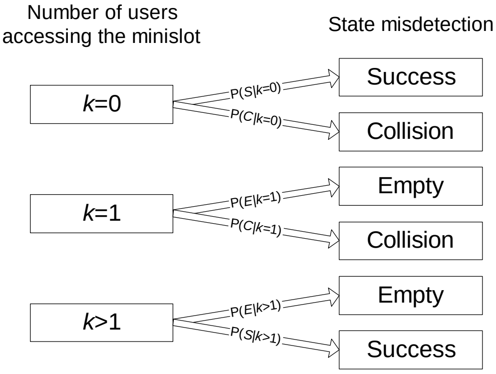

Figure 4. Possible misdetection situations

It is possible to derive analytical expressions for the probability of each error situation as a function of the detection and false alarm probabilities of each branch of the previously presented scheme [13], [14].

As an example, Figure 5 and Figure 6 show the minislot state misdetection probabilities as a function of the receiver filter false alarm and detection probabilities respectively. The presented 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, l=10 and antenna diversity of order D=2.

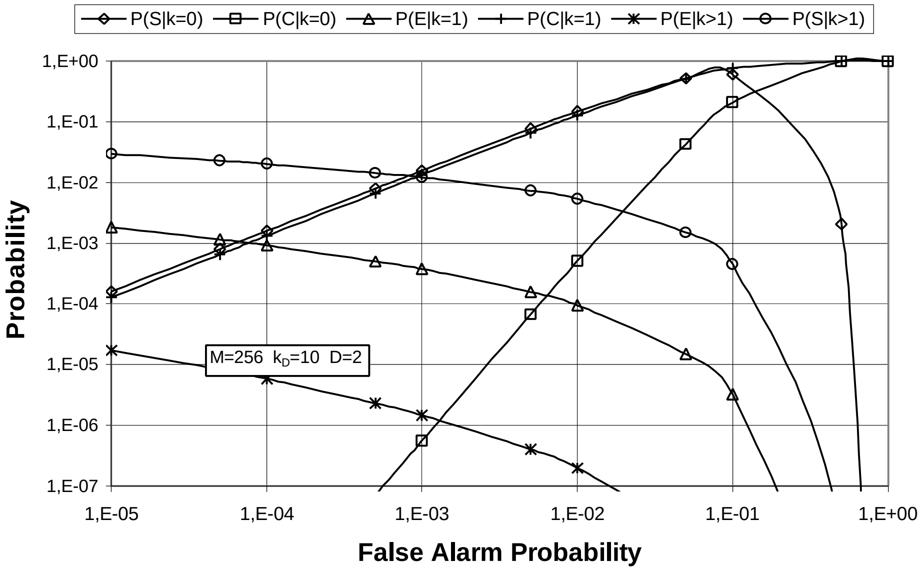

Figure 5. Minislot state misdetection probabilities as a function of P_f

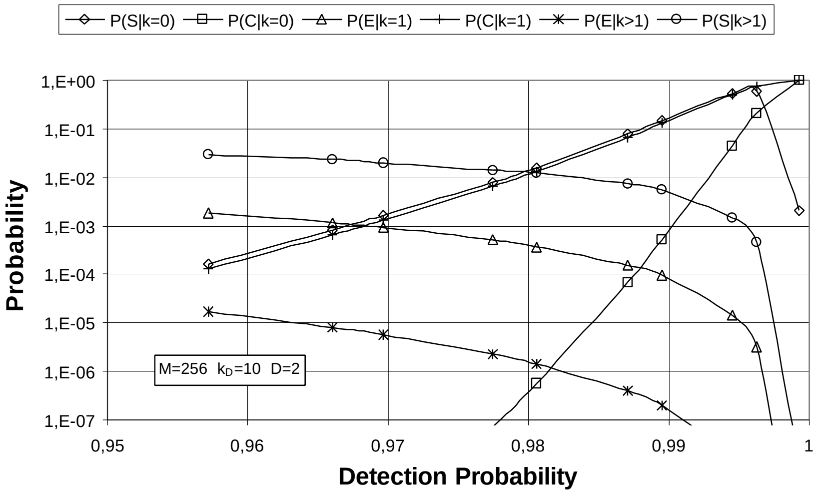

Figure 6. Minislot state misdetection probabilities as a function of P_d

3.4 Protocol modifications for error recovery

Taking into consideration the non-zero probability of the base station failing in the detection of the control minislot states, we must analyse the necessary protocol adaptations to avoid dead-lock situations in a normal data-transmission complete-system steady-state run. The six different error scenarios shown in Figure 4 are analysed 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 2.1 together with changes in the criteria for the base station when sending the broadcast information to the users.

3.5.1 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 with 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 0), 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 minimise the probability of this event, as this number will be the quantity of wasted slots when this error occurs. It would seem to be a matter of a trade-off between efficiency and the probability of a data collision occurring.

3.5.2 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 dead-lock situations.

3.5.3 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 dead-lock situations may appear, only little delay loss is caused and thus no algorithm modifications are needed.

3.5.4 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 dead-lock situation nor entail the need for any protocol modification.

3.5.5 Empty minislot detection/Multiple user transmission ($E|k>1$)

This case is similar to the one presented in 3.5.3 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 dead-lock error is possible.

3.5.6 Success minislot detection/Multiple user transmission ($S|k>1$)

This is the most critical of the possible error situations. 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 non-zero 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 3.5.1 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 maximise 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.

4 SIMULATION RESULTS AND COMPARISONS

Bearing all the ideas presented in mind, computer simulations were carried out to validate the protocol operation. The parameters used in the simulations were N=100 data terminals, average message length of 6400 bits, CDMA channel with K=16 spreading codes 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 for data transmission (not for the control minislots), a single-cell operation environment and used the Gaussian hypothesis for the interference power evaluation in the calculation of the *BLER*.

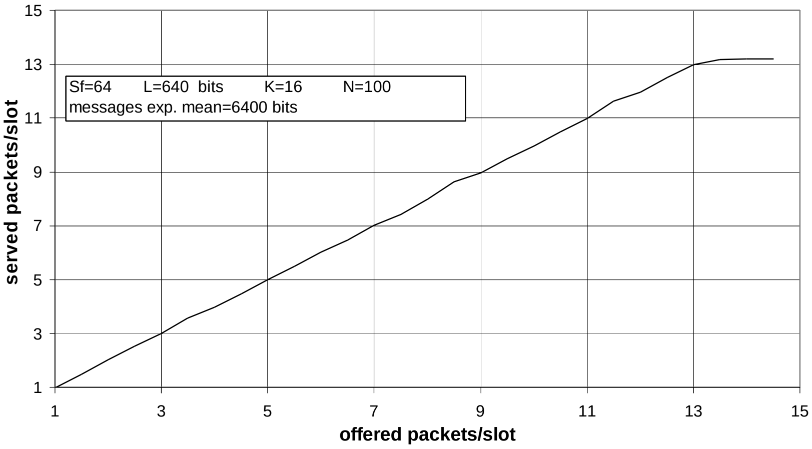

Figure 7. Throughput of DQRAP/CDMA

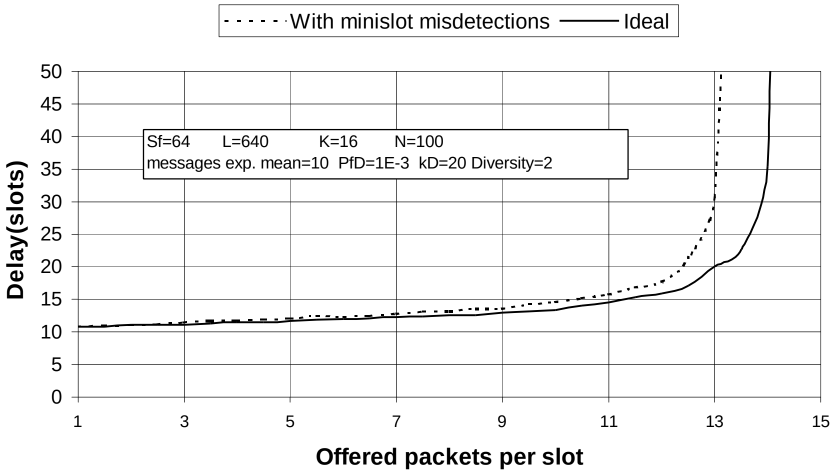

Figure 8. DQRAP/CDMA delay. Ideal vs. possible minislot state misdetection

The simulation results show a remarkable feature of the protocol. The maximum stable throughput achieved by the system is near the optimum (about 13.5L bits/slot for a theoretical maximum channel capacity of 14.3L bits/slot), maintaining a small mean delay for offered traffic up to this maximum value.

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 3.5. Figure 8 shows the delay comparison between the ideal and the non-ideal situation with $P_{fD}=10^{-3}$ and k_D=20 and using antenna diversity of order D=2. We can see the small loss caused by the misdetection in the control minislots and thus the robustness of the protocol.

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.

Finally, we carried out a comparison with other medium access protocols designed for a CDMA environment. Slotted-ALOHA/CDMA [8] has been extensively used as an access scheme for systems where the number of spreading codes is comparable with the number of terminals. 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) [11]. 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.

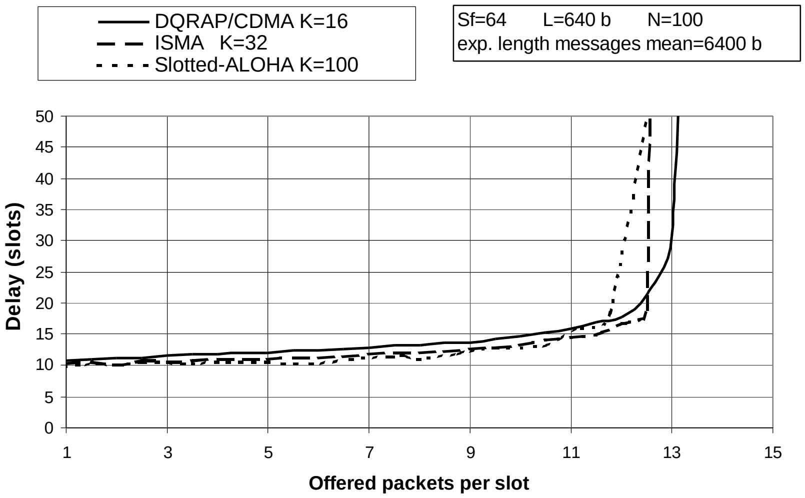

Figure 9. Delay comparison between protocols

Figure 9 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 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 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 loads.

5 CONCLUSIONS

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 minimises the number of spreading codes needed for a particular overall performance. It is therefore a suitable proposal for improving the use of the capacities of random access channels in the reverse link.

A receiver scheme for the detection of access requests has been proposed and analysed, 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, even when using a reduced set of spreading codes.

6 REFERENCES

[1] Dimitri Bertsekas, Robert Gallager, *Data Networks.* Prentice Hall International Editions. 1992.

[2] B.S. Tsybakov, N.B. Likhanov, "Upper Bound on the Capacity of a Random Multiple Access System", Problems of Information Transmission, Vol. 23, No. 3, pp. 224-236.

[3] T. Towsley, P.O. Vales, “Announced Arrival Random Access Protocols”, IEEE Trans. On Communications, Vol. COM-35, No. 5, pp. 513-521, May 1987.

[4] Wenxin Xu, Graham Campbell, "A Near Perfect Stable Random Access Protocol for a Broadcast Channel", IEEE Proceedings of ICC'92, Vol. 1, pp. 0370-0374.

[5] Erik Dahlman, Per Beming, Jens Knutsson, Fredrik Ovesjö, Magnus Persson, Christian Roobol, "WCDMA–The Radio Interface for Future Mobile Multimedia Communications", IEEE Transactions on Vehicular Technology, Vol. 47, No. 4, November 1998.

[6] "UMTS Terrestrial Radio Access: Concept Evaluation (UMTS 30.06)". ETSI Tech. Rep. 101 146, version 3.0.0, December 1997.

[7] Recommendation ITU-R M.1034. "Requirements for the Radio Interface(s) for Future Public Land Mobile Telecommunication Systems (FPLMTS)". 1994.

[8] A. Chockalingam, Weiping Xu, Laurence Milstein, "Performance of a Multi–Channel Packet CDMA Protocol in a Fading Environment", Conference Record, IEEE Vehicular Technology Conference, VTC'97. 1997.

[9] Leonard Kleinrock, *Queueing Systems*, John Wiley and Sons INC., New York, 1976.

[10] M.B. Pursley, “Performance Evaluation for Phase-Coded Spread-Spectrum Multiple-Access Communication – Part I: System Analysis”, IEEE Trans. On Communications, Vol. COM-25, No. 8, pp. 795-799, August 1977.

[11] Jordi Pérez, Ramón Agustí, Oriol Sallent, "Performance Analysis of an ISMA CDMA Packet Data Network", To be published in Proceedings VTC'99 Fall. Amsterdam, September 1999.

[12] M.J. Karol, Z. Liu, K.Y. Eng, "Distributed–Queueing Request Update Multiple Access (DQRUMA) for Wireless Packet (ATM) Networks". Proceedings ICC'95, pp. 1224-1231, Seattle, WA.

[13] Andrew J. Viterbi. *CDMA Principles of Spread Spectrum Communication.* Addison-Wesley Pub. Co. 1995.

[14] L.G. Alonso, R. Agustí, O. Sallent, “A Near-Optimum Medium Access Control (MAC) Protocol Based on the Distributed Queueing Random Access Protocol (DQRAP) for a CDMA Mobile Communication System”, submitted to IEEE Journal on Selected Areas in Communications, July 1999.