PROPOSAL OF DQRAP/CDMA MAC PROTOCOL OPTIMIZATION

Giulio Giovannini, Luis Alonso, Ramón Agustí
Dept. Signal Theory and Communications. Polytechnic University of Catalonia (UPC)
c) Jordi Girona 1-3. Campus Nord. Edifici D4. 08034.-Barcelona. Spain
e-mail: ggiov@nortelnetworks.com, luisg@tsc.upc.es, ramon@tsc.upc.es

Abstract – Future third-generation mobile communication systems will need MAC protocols suitable for multimedia CDMA radio communications. DQRAP/CDMA is a general purpose MAC protocol oriented to the CDMA environment. Analytical model expressions and computer simulations have shown its capacity to achieve near-optimum performance under certain traffic scenarios. Starting from the DQRAP/CDMA specification presented in [1], we propose some algorithm modifications in order to further improve the system performance when the traffic load is extremely bursty. Computer simulations have been carried out to evaluate this performance improvement.

1 INTRODUCTION

In the last few years, many MAC protocols suitable for multimedia CDMA radio communications have been proposed. The first ones being studied and used were random access protocols, like ALOHA and Slotted-ALOHA. They present good delay performance for light traffic load. However, throughput instability appears when the traffic load increases. Collision Resolution Algorithms (CRA) were introduced to achieve better throughput performance, but, in general, the delay performance is affected by the required access procedure.

DQRAP/CDMA [1] has shown its capacity to achieve near-optimum performance under certain traffic scenarios. The most interesting feature of DQRAP/CDMA is its ability to combine the good delay performance of random access protocols and the good throughput performance of reservation protocols. In fact, it behaves like a random access protocol when the traffic load is light and switches smoothly and automatically to a reservation protocol when the traffic load increases. The goal is achieved by the introduction of two distributed queues. These queues schedule user's system access requests and data information transmissions. Moreover, these queues are able to collect the traffic load that exceeds the system capacity preventing the typical channel interference overload of random access protocols.

Analytical expressions and computer simulations presented in [1] show the good delay and throughput performance of DQRAP/CDMA. It is able to approach the ideal M/M/K queueing system performance. Moreover, the results about the delay standard deviation highlight the fair behaviour of the protocol that maintains to a minimum the *jitter* of the packet delay.

In this paper, we propose some modifications to the algorithm rules presented in [1], in order to further improve the performance of the system even for the worst case traffic scenario. The proposed modifications are designed to have a minimum impact on the other layers of the system which result unaltered.

This paper is organized as follows. The framework is described in section 2. In section 3 we outline the protocol operation and performance. In section 4 we describe the proposed modifications and, in section 5, computer simulations results are presented, showing the improved performance of the modified protocol. Finally, section 6 is devoted to the conclusions.

2 FRAMEWORK

We consider a system consisting of N users that share a common CDMA radio channel to communicate with a base station. Uplink and downlink channels are supposed to be always perfectly synchronized.

Time axis is divided into slots. Data transmissions are organized in packets with fixed duration of one slot.

The number of spreading codes being used in the system is C. The spreading codes are ordered and C_i represents the i-th code. The users are supposed to be able to change the spreading code on a slot-by-slot basis.

We assume that an ideal closed loop power control is implemented for the data transmission. Using the Gaussian hypothesis for the user reciprocal interference and neglecting the thermal noise, the Bit Error Probability (*BER*) is approximated by [2]:

$$BER = \frac{1}{2} erfc\left(\sqrt{\frac{3S_f}{2(n-1)}}\right) \quad (1)$$

where S_f is the spreading factor used for transmission and n is the number of simultaneous transmissions.

If we consider correctly transmitted only those packets which carry no erroneous bits, the probability to transmit an incorrect packet (*BLER*) is [1]:

$$BLER = 1-(1-BER)^{L_p} = 1-\left[1-\frac{1}{2} erfc\left(\sqrt{\frac{3S_f}{2(n-1)}}\right)\right]^{L_p} \quad (2)$$

where L_p is the packet length expressed in bits.

The average packet effective throughput, expressed in packets sent without errors per slot, is then:

$$S = n(1-BLER) = n\left[1-\frac{1}{2} erfc\left(\sqrt{\frac{3S_f}{2(n-1)}}\right)\right]^{L_p} \quad (3)$$

3 PROTOCOL OVERVIEW

The DQRAP/CDMA time structure is shown if figure 1.

Time is divided into slots that are further divided into two subslots. They are the data subslot, devoted to the data packet transmission, and the control subslot, devoted to the transmission of user access requests. The control subslot is further divided into m access minislots that are ordered following a time criterium. The protocol also requires that along with each data packet, users transmit a *final message bit* that is set to ON only for packets that are at the end of a message.

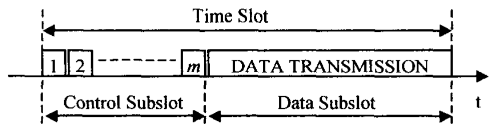

Figure 1. Time axis structure

The basic elements introduced by the DQRAP/CDMA protocol are two distributed queues. They are the *Data Transmission Queue* (DTQ), devoted to the data packet transmission scheduling, and the *Collision Resolution Queue* (CRQ), devoted to the collision resolution algorithm.

These queues are represented by four integer numbers. This values must be maintained and updated every slot by each user. We will call them TQ, RQ, pTQ and pRQ. TQ is the number of messages waiting for transmission in the DTQ. RQ is the number of collisions waiting for resolution in the CRQ. pTQ is the user position within the DTQ, while pRQ is the user position within the CRQ.

We remark that TQ and RQ have always the same value for all users (i.e. they are "distributed" queues) while pTQ and pRQ may differ between users as they denote the positions within the queues of each one of them.

The main idea of DQRAP is to concentrate user accesses and collisions in the control subslots while the data subslots are devoted to collision-free data transmission.

Let us consider a user that has just arrived to the system and has data to transmit. It will apply the following access procedure:

- It checks the state of both the *distributed queues* in order to decide whether it is enabled to attempt a system access. This is a key feature of the protocol as avoids unstable situations. In fact, users will be forbidden from attempting accesses if the system is already working at the maximum of its capacity. If the user is enabled, it selects one of the C available spreading codes applying a certain set of rules defined by the DQRAP/CDMA protocol [1]. Let us call C_s the selected spreading code.
- It randomly selects one of the m control minislots. Let us call m_s the selected minislot.
- It transmit an access request in minislot m_s using the spreading code C_s.

After the access request transmission, two options are possible:

- No other user has transmitted an access request using the same m_s and C_s. In this case the access request is *successful* and the user enters the DTQ. In the DTQ it will wait for its turn to transmit a data packet and it will be inhibited from sending new access requests.
- Two or more users have transmitted access requests using the same m_s and C_s. In this case the access request has *collided* and the affected users enter the CRQ. In the CRQ they will wait for their turn to transmit a new access request in order to enter the DTQ.

The DQRAP/CDMA algorithm is defined by three set of rules that each user has to perform at the end of every slot. For details on these rules refer to [1].

We will now present some computer simulation results that show the protocol performance. We have considered the case of N=100 users that generate a Poisson-type traffic. The message length is a exponentially distributed random variable with mean L_m packets/message. The spreading factor used for transmission is S_f=64 and the packet length is L_p=600 bits. The number of spreading codes is C=17 that correspond to the optimum number of simultaneous transmissions that maximize the packet throughput for the chosen spreading factor when we do not consider any channel coding to correct erroneous bits [1]. The number of minislots per spreading code is set to m=3. The packet error probability is given by expression (3) while for the minislot reception we have used the reception scheme detailed in [1]. It is assumed that the minislot access requests transmissions are under an open loop power control, and thus they suffer from Rayleigh fading conditions.

In figure 2 we report the average message delay normalized to the message average length for different values of this latter average (the line marked as L_d=1 corresponds to a case of deterministic message length always equal to 1 packet/message).

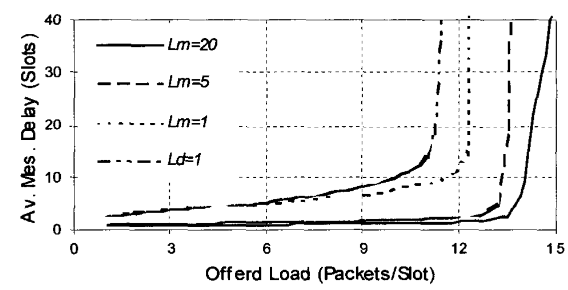

Figure 2. Average message delay normalized to the average message length

For light traffic load the protocol is able to maintain the average message delay close to the value that corresponds to the one needed to transmit the message. The best performance is obtained for long messages (L_m=20) while, the average message delay results higher when the average message length decreases. The shorter the message length, the higher the number of access requests that arrive each slot to the minislots for the same total traffic load. In fact, for L_m=1 and L_d=1 users transmit quickly their messages and attempt more frequently to access the system. This fact causes an increase of the number of collisions and thus users enter more frequently the CRQ.

In figure 3 we present the average packet throughput for the same conditions of figure 2.

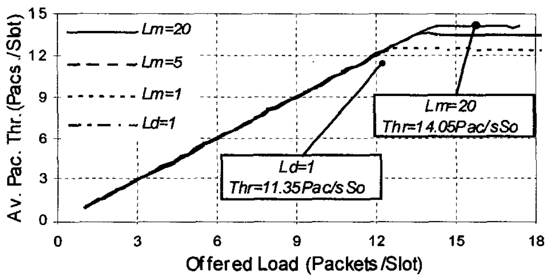

Figure 3. Average packet throughput.

For light traffic load, the average packet throughput follows closely the offered load. For high traffic load the average message length sets differences in this performance. In the case of long messages the system is able to correctly deliver 14.05 packets per slot that correspond to the 96.95% of the maximum theoretical throughput. In fact, from expression (3) we derive that the maximum deliverable throughput is S_{th}=14.49 packets/slot for the chosen values of S_f and L_p [1]. For short messages (L_d=1) the maximum throughput achieved is 11.35 packets/slot that it is only the 78% of S_{th}. We further observe that the system throughput is kept stable even when the offered traffic gets over the maximum sustainable. This feature is achieved thanks to the fact that the exceeding load that could provoke instability situations is maintained in the DTQ and do not cause a non desirable extra interference level.

4 PROTOCOL OPTIMIZATION

We will now propose two modifications to the Request Transmission Rules (RTR) [1] set of rules in order to improve the system performance.

Looking at this rules, we observe that RTR-2 performs a biunivocal association between the first C positions of the CRQ and the C available spreading codes. In other words, only users that in slot i are found in a position that accomplishes pRQ<C have access to the minislots of the spreading code C_{pRQ}. In this way, users have m minislots to resolve their collisions. We further observe that if in slot i we have RQ<C, the minislots of the last C–RQ spreading codes will not be used by users in the CRQ. RTR-1 takes advantage of this facts assigning the first unused spreading code, that is the spreading code C_{RQ+1}, to the *new users* accesses. We here define *new user* as a user that is found out of the *distributed queues* (that corresponds to pTQ=0 and pRQ=0). Therefore, *new users* will have m available minislots for their access attempt. The spreading code assignation we just have described is illustrated in figure 4.

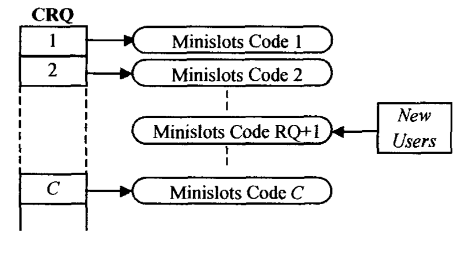

Figure 4. Standard Code Assignation

On the base of the above observation we propose the following algorithm modifications:

Modification to RTR-1: we propose to reserve the last C_r (with C_r>1) spreading codes for *new user* access requests. This means that the last C_r spreading codes are always available for the first access attempts of *new users*, independently of the CRQ occupation. Moreover, if C–RQ>C_r, we propose to assign all the last C–RQ spreading codes for *new users* access (figure 5). In this way, *new users* have at least mC_r (with mC_r>m) minislots for their first access attempt.

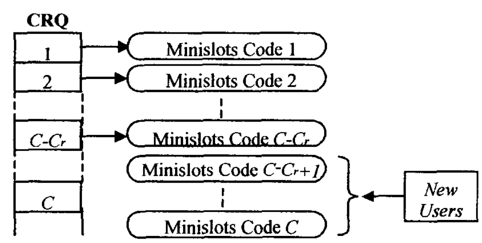

Figure 5. Modified RTR.1

The modified version of RTR.1 is:

- RTR-1: if a user has pTQ=0 and pRQ=0 (*new user*), it randomly selects a spreading code in the range between $C_{min(RQ,C-Cr+1)}$ and C_C. Afterwards it randomly selects a minislot and transmits an access request in it using the selected code.

Modification to RTR-2: we propose to assign C_a (with C_a>1) spreading codes to each position of the CRQ (figure 6). In this way users have mC_a (with mC_a>m) minislots to resolve their collisions. The modified RTR-2 is:

- RTR-2: if a user has 0<pRQ≤$\lfloor (C - C_r)/C_a \rfloor$, it randomly selects a spreading code in the range between $C_{(pRQ-1)Ca+1}$ and $C_{pRQ \cdot Ca}$. Afterwards it randomly selects a minislots and transmits and access request in it using the selected spreading code.

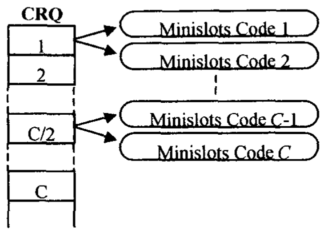

Figure 6. Modified RTR.2

5 MODIFIED PROTOCOL PERFORMANCE

We present some results obtained from computer simulations introducing both described modifications. The values the parameters being used are the same of figures 2 and 3 with the exception of the average message length that is chosen to be always deterministic and equal to 1 packet/message (case of L_d=1). This case represents a worst case scenario (figures 2 and 3) as it loads the system with the most bursty possible traffic load. We report graphics about the maximum average packet throughput with bounded message delay as it is an indicator of the general system performance.

In figure 7, the maximum average packet throughput as a function of C_r is shown, for different values of C_a.

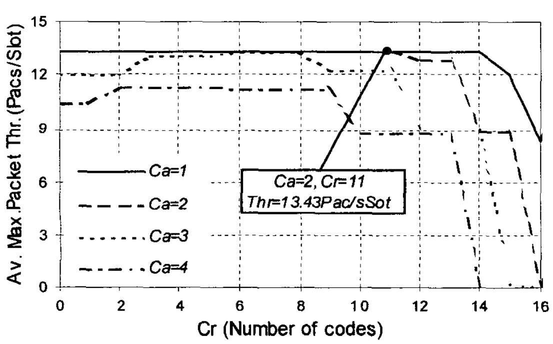

Figure 7. Average Packet Throughput vs. *Cr*

We observe that the best results are obtained with the couple of values C_a=2 and C_r=11 for which the throughput reaches the value of 13.43packets/slot (92.67% of S_{th}). Therefore, it is shown that, thanks to the optimisation process, we have improved the maximum throughput in about 2 packets/slot with respect to the original protocol (11.35 packets/slot, 78,32% of S_{th}, with L_d=1).

In figure 7 we have optimized the system performance for one specific value of m (m=3). In figure 8 we present the maximum average packet throughput when we optimize C_a and C_r for different values of m. The maximum throughput is now expressed as a percentage of the maximum theoretical throughput, S_{th}.

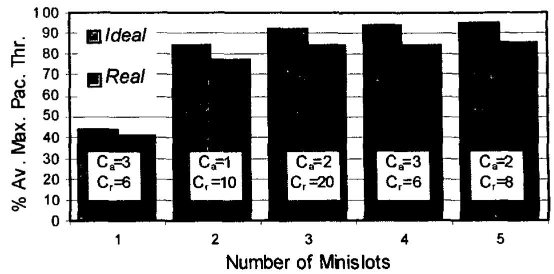

Figure 8. Optimized averaged packet throughput for different values of *m* (expressed as % of S_{th})

Moreover, in figure 8 we report the values of this throughput when taking into account the losses caused by the non-zero duration of the access minislots and the error detection bits added to each data packet. We will refer to this second throughput evaluation as *effective throughput*. To evaluate this effective throughput, the slot duration has been chosen to be equivalent to the UTRA (UMTS Terrestrial Radio Access[5]) CPCH frame duration, that is T_s=10 ms, while the minislot duration has conservatively been chosen as double the time strictly needed to transmit an access request of 256 chips, using the UTRA chip rate specification (R_c=3.84Mchips/slot [5]). With this value, the minislot duration results to be T_m=0.1334 ms (1.334% of T_s). Finally, it is assumed that the error detection code consists in 32 CRC bits per packet.

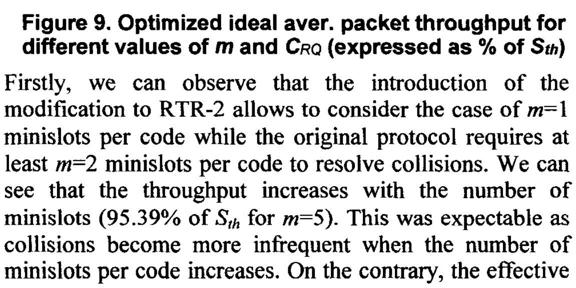

Figure 9. Optimized ideal aver. packet throughput for different values of *m* and C_{RQ} (expressed as % of S_{th})

Firstly, we can observe that the introduction of the modification to RTR-2 allows to consider the case of m=1 minislots per code while the original protocol requires at least m=2 minislots per code to resolve collisions. We can see that the throughput increases with the number of minislots (95.39% of S_{th} for m=5). This was expectable as collisions become more infrequent when the number of minislots per code increases. On the contrary, the effective throughput reaches a maximum for m=3 minislots per code (84.38% of S_{th}) and does not increase for bigger values of m. This behaviour is reasonable as bigger numbers of minislots per code result in higher duration of the control subslots and therefore in higher overhead losses.

As a final step in the optimization work we have considered variation in the number of available spreading codes used during the control subslot. In fact, the time division multiplexing that exists between the control subslot and the data subslot permits to adopt different numbers of maximum simultaneous transmissions for each of them. Let us call C_{RQ} the number of spreading codes being used during the control subslot.

Figure 9 presents the maximum ideal average packet throughput obtained when we optimize the three parameters C_a, C_r and C_{RQ}. We have considered two different traffic situations. The first corresponds to short messages of deterministic length (L_d=1) while the second to long exponentially distributed messages (L_m=20).

The throughput performance results improved for all the values of m. As an example, in the case of L_d=1 packet/message and m=3 minislots per slot, the maximum average throughput reaches the 97.72% of S_{th} (in the previous case, for m=5, the throughput obtained was 95.39% of S_{th}). When L_m=20 packets/message, which represents a lighter traffic load in terms of access requests, m sets little differences as the 98% of S_{th} is always achieved even in the case of m=1 minislot per code. Therefore, in this latter case of long messages, the modified protocol using only one minislot per slot outperforms the original protocol using m=3 minislot per code (see figure 2).

In figure 10 we present the effective throughput obtained under the same conditions of figure 9.

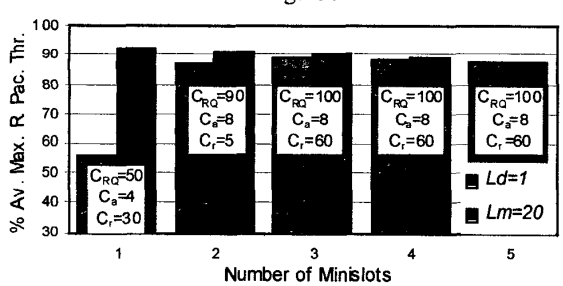

Figure 10. Optimized effective aver. packet throughput for different values of m and C_{RQ} (expressed as % of

In the case of short messages (L_d=1) the optimum values are found for m=3 (88.95% of S_{th}). On the contrary, in the case of long messages, the best performance is obtained for only one minislot per code (91.99% of S_{th}) which minimizes the losses introduced by the minislots presence. Therefore we may conclude that the optimum value of m depends on the type of traffic received by the system. Anyway, an intermediate choice of m=2 seems a good trade-off option which achieves good overall performance under all the considered traffic situations.

To better evaluate the improvement achieved by the protocol optimization we present in figure 11 a comparison between the packet delay cumulative probability performance of the original and optimized protocol. The packet delay cumulative probability gives the probability a packet has to be correctly delivered within a given delay. The curves have been obtained for the case of m=3 minislots per code, with messages of deterministic length equal to L_d=1 packets/message and for an offered load of 10 packets/slot.

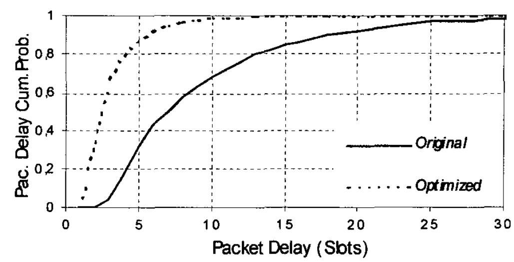

Figure 11. Pac. delay cumulative probability comparison

We observe that the optimized protocol outperforms significantly the original protocol. As an example, the optimized protocol delivers 87% of the packets within 5 slots while the original protocol delivers only 31% of the packets within the same delay.

6 CONCLUSIONS

Modifications to the *Request Transmission Rules* (RTR) of the DQRAP/CDMA MAC protocol have been proposed in order to improve the overall throughput performance for the worst traffic load scenario. Simulation results show the performance improvement achieved by the modified protocol. For the case of very short frequent messages, the throughput performance is improved in about a 20%. For the case of long messages the modified system is able to achieve near-optimum performance with as little as only one minislot per code, keeping an average maximum throughput over the 98% of the maximum theoretical one. These results are very promising and show a good potential capacity of the protocol for packet switched wireless communications.

7 REFERENCES

[1] Luis G. Alonso, Ramón Agustí, Oriol Sallent, "A Near-Optimum MAC Protocol Based on the Distributed Queueing Random Access Protocol (DQRAP) for a CDMA Third Generation Mobile Communication System", IEEE Journal on Selected Areas in Communications, Vol.18, No.9, pp.1701-1718, sept 2000.

[2] Andrew J. Viterbi. *CDMA Principles of Spreading Spectrum Communication*, Addison-Wesley Pub. *Collision*. 1995.