A NEAR-OPTIMUM MEDIUM ACCESS PROTOCOL BASED ON THE DISTRIBUTED QUEUEING RANDOM ACCESS PROTOCOL (DQRAP) FOR A CDMA THIRD GENERATION MOBILE COMMUNICATION SYSTEM

Luis G. Alonso, Ramón Agustí, Oriol Sallent
Dept. Signal Theory and Communications.Polytechnic University of Catalonia (UPC)
c) Jordi Girona 1-3. Campus Nord. Edifici D4. 08034.-Barcelona. Spain
e-mail: [luisg, ramon, oriol]@tsc.upc.es

***Abstract*[*]-**This paper presents and analyses a new near–optimum random access protocol. The proposed access scheme is suitable for a CDMA environment and minimises the total number of spreading codes needed to achieve a certain throughput. It is based on distributed queues and a collision resolution algorithm. Computer simulations have been carried out to validate the analytical model and the obtained results show that the protocol has very good stability and delay performance, even compared with medium access protocols using more available spreading codes.

1 INTRODUCTION

In the past few years, many research efforts have been focused in the design of Media 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 the flexibility and efficiency enough to allow the existence of this kind of systems with a reasonable complexity and reliability.

ALOHA and Slotted–ALOHA techniques have been largely 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 originated the appearance of collision resolution algorithms (CRA), also called tree algorithms [1], which have a higher performance. One of the best–performing MAC protocols based on collision resolution proposed to date is DQRAP [2], a protocol initially designed for a TDMA environment. 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 even higher than unity.

On the other side, Direct-Sequence Code Division Multiple Access (DS-CDMA) is one of the most likely candidates for the 3rd generation mobile telecommunication systems. Schemes based on Wide-band CDMA (WCDMA) [3] have been chosen as radio interface in the standardisation body in Japan (ARIB), and also in Europe by the ETSI for the UMTS Terrestrial Radio Access (UTRA) [4]. This access scheme is also being considered in the International Mobile Telecommunication 2000 (IMT-2000) [5] by the ITU. In this paper, we propose a random near–optimum medium access protocol that modifies and extends DQRAP techniques for being used in a CDMA environment as those mentioned above. The great performance of this 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 for efficiently transmitting data.

DQRAP is based on two distributed queues that every terminal has to maintain updated based on a simple feedback information broadcasted by the base station. It also uses a collision resolution algorithm that avoids instability by, in average, resolving a collision of *n* packets in less time that the one the system needs to transmit them. This is achieved by using only as few as three control minislots per slot.

In the CDMA environment, we introduce the idea of using a DQRAP engine for each one of the spreading codes, and then unifying the collision resolution and transmission queues corresponding to each spreading code 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. We also introduce a modification that allows the terminals to transmit consecutively the packets pertaining to the same message without losing the simplicity of the control minislots (they remain carrying no information). This last feature reduces at minimum the *jitter* of the delay of the packets corresponding to one message and becomes also a new advantage to manage messages of more-than-one-slot length.

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. This behaviour is the key of its good delay and throughput performance. The main advantage of this protocol in a CDMA environment is that, for a given channel characteristics, minimises the number of spreading codes needed to reach a certain throughput. So it will be possible to use a very good set of spreading

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

sequences, thus reducing the total interference level and allowing to increase the system capacity. In CDMA, power is the resource to be shared. This protocol allows to decide the power used by each of the terminals and know the maximum total interference level, avoiding the system from generating more interference than the one required for a certain overall performance.

Furthermore, a comparison with a Slotted-ALOHA/CDMA protocol has been carried out, showing that DQRAP/CDMA outperforms in about 15% the maximum stable throughput of this protocol, even having one spreading code assigned to each terminal (no code collisions) in the Slotted-ALOHA system, while having only 16 codes for 100 users in DQRAP/CDMA. This shows up a dramatic increase in terms of absolute throughput.

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 derived also in this section. Section 4 shows computer simulation results and comparisons with other protocols. Finally, section 5 is devoted to the conclusions.

2 PROTOCOL DESCRIPTION

Let consider *N* data terminals who 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 parts. The first part is the access part which is further divided into three control minislots. The second part 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. Data terminals generate Poisson-distributed messages of exponential-distributed length with mean ($1/m$). We consider that the terminals are able to change the spreading code for data and request transmission every slot. The messages generated by one terminal are splitted into slot-duration packets and put in a buffer. Each packet will be sent with the same spreading code but not all the packets pertaining to one message will be necessarily 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 to the system, the corresponding terminal sends a request in one of the control minislots. If it fails (the request collides with one or more requests from other messages), it enters the collision resolution queue. The transmission queue contains the messages that have succeeded in their request and are waiting to be transmitted to the base station.

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 being resolved in the distributed collision resolution queue. pTQ is the position of the terminal in the data transmission queue and pRQ is the position of the terminal in the collision resolution queue. All queues are FIFO. This values are initially set to zero and must be kept updated using the feedback information sent by the base station and following the set of rules described below. This feedback information must be sent to all the terminals every slot using a broadcast channel. It consists of a ternary state data for each control minislot of every spreading code, and has also to include a final-message-bit for every code. The three different states 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. We will consider an ideal broadcast feedback channel.

The protocol algorithm consists in three sets of rules that each data terminal has to follow at the end of every 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 describe now the algorithm rules that each data terminal has to execute at the end of every slot, assuming that, at this moment, 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 same order that are presented.

2.1.1 QDR (Queueing Discipline Rules)

1. Each station increments the value of TQ in one unit for each control minislot that has the success state, taking into account all the control minislots from any of the *K* spreading codes.
2. Each station reduces the value of TQ in 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 in min(RQ, *K*) units.
4. Each station increments the value of RQ in one unit for each control minislot that has 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 succeed, calculates its position between all the control minislots with this state 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 time arrival criteria. On the other side, if the request has collided, the terminal calculates its position

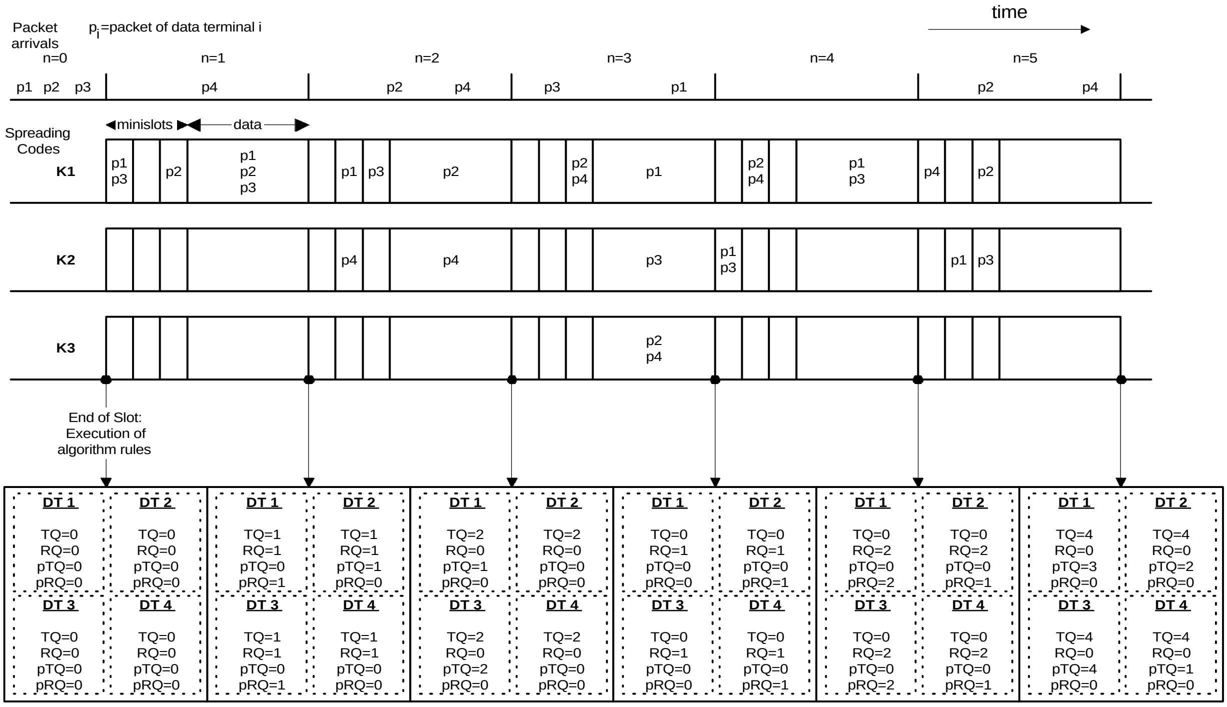
between all the collisions and sets pRQ to the corresponding value at the end of RQ. If it has not sent any request, then pTQ and pRQ follow the same update rules as TQ and RQ respectively, but only if the initial values are different from zero.

2.1.2 DTR (Data Transmission Rules)

1. If TQ<K and pTQ=0 and pRQ=0, each station that has data packets ready to be sent transmits the first packet of its buffer using the spreading code $K_{\mathrm{TQ+1}}$.
2. If pTQ>0 and pTQ≤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 to ON the final-message-bit.

2.1.3 RTR (Request Transmission Rules)

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



DT 1	DT 2	DT 1	DT 2	DT 1	DT 2	DT 1	DT 2	DT 1	DT 2	DT 1	DT 2
TQ=0 RQ=0 pTQ=0 pRQ=0	TQ=0 RQ=0 pTQ=0 pRQ=0	TQ=1 RQ=1 pTQ=0 pRQ=1	TQ=1 RQ=1 pTQ=1 pRQ=0	TQ=2 RQ=0 pTQ=1 pRQ=0	TQ=2 RQ=0 pTQ=0 pRQ=0	TQ=0 RQ=1 pTQ=0 pRQ=0	TQ=0 RQ=1 pTQ=0 pRQ=1	TQ=0 RQ=2 pTQ=0 pRQ=2	TQ=0 RQ=2 pTQ=0 pRQ=1	TQ=4 RQ=0 pTQ=3 pRQ=0	TQ=4 RQ=0 pTQ=2 pRQ=0
DT 3	DT 4	DT 3	DT 4	DT 3	DT 4	DT 3	DT 4	DT 3	DT 4	DT 3	DT 4
TQ=0 RQ=0 pTQ=0 pRQ=0	TQ=0 RQ=0 pTQ=0 pRQ=0	TQ=1 RQ=1 pTQ=0 pRQ=1	TQ=1 RQ=1 pTQ=0 pRQ=0	TQ=2 RQ=0 pTQ=2 pRQ=0	TQ=2 RQ=0 pTQ=0 pRQ=0	TQ=0 RQ=1 pTQ=0 pRQ=0	TQ=0 RQ=1 pTQ=0 pRQ=1	TQ=0 RQ=2 pTQ=0 pRQ=2	TQ=0 RQ=2 pTQ=0 pRQ=1	TQ=4 RQ=0 pTQ=4 pRQ=0	TQ=4 RQ=0 pTQ=1 pRQ=0

Figure 1. Example of DQRAP/CDMA protocol operation

2.2 Example

The example shown in Figure 1 illustrates the operation of the protocol with K=3 and N=4. All the messages generated by the terminals are assumed of length one, so every data slot has the final-message-bit set to on.

In the slot n=0 three messages arrive to the system. In n=1 they try to send a request and also to transmit the data in the first spreading code. Only *p2* request succeeds and *p2* enters the transmission queue. As *p1* and *p3* requests collide, they enter the collision resolution queue. Data packets also collide. In this slot a message from *p4* arrives to the system. In n=2, *p2* is transmitted using the first spreading code. Packets *p1* and *p3* resolve their collision and enter the transmission queue (*p1* in the first position as its request used a prior control minislot). On the other side, *p4* transmits its request and data using the second spreading code. As *p4* is the only new packet arriving to the system, its data transmission succeeds and then it does not need to enter any queue. Two more packets arrive in this slot.

In n=3, *p1* and *p3* are transmitted using the first and second spreading codes. The new *p2* and *p4* packets send their requests and collide. They enter the collision resolution queue. In n=4, requests from *p2* and *p4* collide and the packets enter again 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 try 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) Queue represents the time each message has to wait since it arrives to the system until the next time slot starts. Both subsystems have as many servers as available spreading codes (i.e. *K*). We will see in section 3.1 that the service time of both subsystems is exponential distributed, therefore they are two M/M/*K* systems.

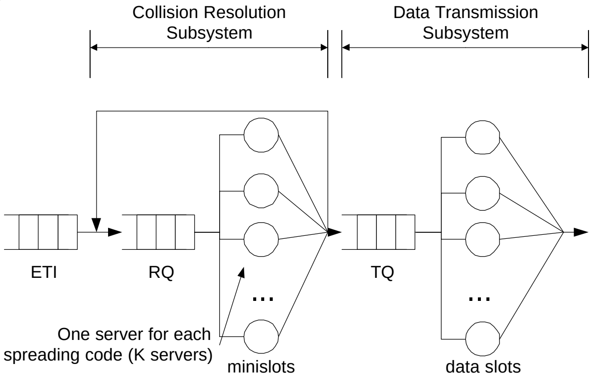

Collision Resolution Subsystem
Data Transmission Subsystem
ETI
RQ
TQ
One server for each spreading code (K servers)
minislots
data slots
...
...

Figure 2. Model of DQRAP/CDMA protocol

It is shown in [1] that, for a TDMA slotted channel, with only three control minislots the average speed of the contention resolution subsystem is greater than the data transmission rate, so the whole system is only limited by the data transmission subsystem. As we have three control minislots per spreading code, the same property still holds. Therefore, the only stability condition is that the input traffic has to be less or equal the total server capacity for the data transmission subsystem, that is, the total transmission rate. Simulation results will confirm this issue.

3.1 Delay Analysis

The total service time for a message (t_{T}) can be decomposed in four terms: the service time of the ETI (t_{ETI}), the service time of the collision resolution subsystem (t_{RQ}), the service time of the data transmission subsystem (t_{TQ}) and the delay produced 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 the rule 1 of the DTR in the same slot. So we can write:

$$t_{T}=t_{ETI}+t_{RQ}+t_{c}+t_{TQ} \qquad (1)$$

If we take the expected value, averaging for all the messages:

$$E[t_{T}]=E[t_{ETI}]+E[t_{RQ}]+E[t_{c}]+E[t_{TQ}] \qquad (2)$$

With this expression we can evaluate the total average delay of the system. Now we describe the expression of the terms in (2). The residual time of the ETI, E[t_{ETI}], equals 0.5 because the arrival of messages is independent of the slot synchronisation.

If l is the total message input rate to the system (with Poisson distribution), it can be proved that the service time of the collision resolution subsystem is also Poisson distributed with mean:

$$\frac{1}{m_{RQ}}=\left[\ln\left(\frac{1}{1-P(l)}\right)\right]^{-1} \qquad (3)$$

where $P(l)$ is the probability that a message will find a control minislot free to access when arrives to the system. If we have *m* control minislots per code, this probability is:

$$P(l)=e^{-l}+\sum_{n=1}^{\infty}\frac{l^{n}}{n!}e^{-l}\left(1-\frac{1}{m}\right)^{n}=e^{-\frac{l}{m}} \qquad (4)$$

Following the M/M/*K* analysis [6], and adding the waiting time in the queue plus the service time, we can write the total delay for the collision resolution subsystem:

$$E[t_{RQ}]=\frac{1}{\ln\left(\frac{1}{1-e^{-l/m}}\right)}+\frac{P_{KRQ}}{K\ln\left(\frac{1}{1-e^{-l/m}}\right)(1-r_{RQ})} \qquad (5)$$

where

$$r_{RQ}=\frac{l}{Km_{RQ}} \qquad (6)$$

$$P_{KRQ}=\frac{\dfrac{(Kr_{RQ})^{K}}{K!(1-r_{RQ})}}{\sum_{n=0}^{K-1}\dfrac{(Kr_{RQ})^{n}}{n!}+\dfrac{(Kr_{RQ})^{K}}{K!(1-r_{RQ})}} \qquad (7)$$

This last expression is the Erlang C formula for the delay probability. Note that in our framework, *m*=3.

As both arrival and service time processes are Poisson distributed, the collision resolution subsystem output traffic pattern will be also Poisson, and, as shown in [7], with the same rate as the input traffic l. Therefore, the total delay for the transmission subsystem has the same terms as for the collision resolution subsystem but changing the service time rate m_{RQ} by a new value m_{TQ}. This value equals to:

$$\frac{1}{m_{TQ}}=\frac{1}{m(1-BLER)} \qquad (8)$$

where *BLER* is the block error probability (the probability that a data packet has at least one error bit) and 1/*m* is the mean length of the messages generated by the data terminals. It is assumed that the system has an error detection system and a Stop & Wait ARQ strategy.

When evaluating the new r_{TQ} in the total delay expression for the data transmission subsystem, we must also substitute the value of m_{RQ} by the new value m_{TQ}.

The only possible situation where a data collision can occur is when the system has less than *K* messages waiting in the data transmission subsystem and more than one packet arrives to the system. The mean delay produced by this fact will be the probability of this event, since if a data collision occurs the message will enter any of the two subsystems of the model and will not collide any more. We can evaluate this probability as:

$$E[t_{c}]=\frac{\sum_{n=0}^{K-1}\dfrac{(Kr_{TQ})^{n}}{n!}}{\sum_{n=0}^{K-1}\dfrac{(Kr_{TQ})^{n}}{n!}+\dfrac{(Kr_{TQ})^{K}}{K!(1-r_{TQ})}}\left(1-e^{-l}(1+l)\right) \qquad (9)$$

4 SIMULATIONS AND COMPARISONS

Figure 3 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 and a spreading factor S_f=64. The slots are of *L*=640 bits. We have used the gaussian hypothesis for the interference power evaluation and assumed a perfect power control. Figure 4 shows the standard deviation of the message delay.

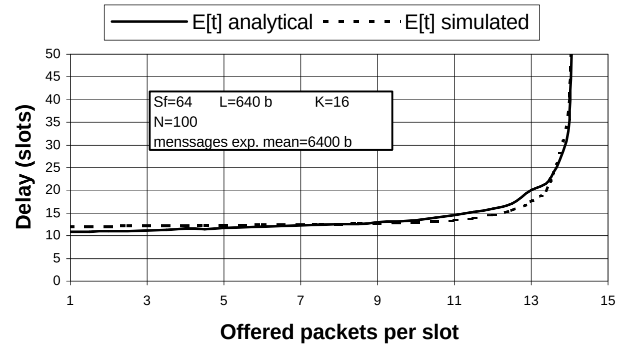


Figure 3. Analytical and Simulation results for the protocol

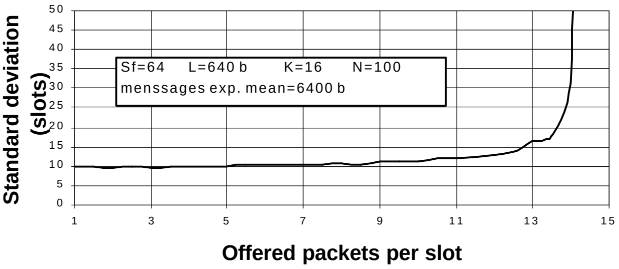


Figure 4. Standard deviation of the message delay

Simulation results fit correctly with the analytical model and show a very important feature of the protocol, that is, the standard deviation of the delay is low bounded.

Finally, we have carried out a comparison with another random access protocol designed for a CDMA environment. Slotted-ALOHA/CDMA has been extendedly 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, makes the transmission immediately.

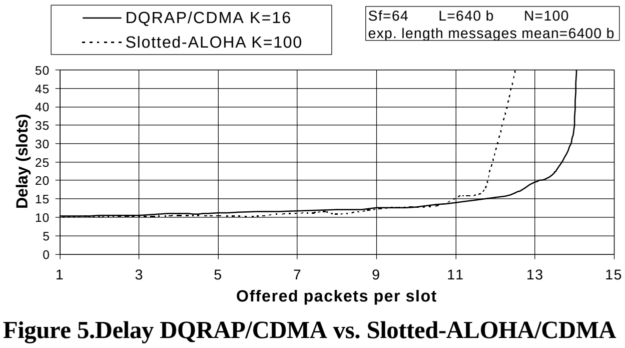


Figure 5.Delay DQRAP/CDMA vs. Slotted-ALOHA/CDMA

Figure 5 shows the delay comparison between a Slotted-ALOHA system with *N*=100 users, where each terminal has an assigned spreading code, and a system using DQRAP/CDMA protocol with the same number of users but using only *K*=16 available spreading codes. Channel conditions are the same for both systems (spreading factor S_f=64, slots of length *L*=640 bits and messages of exponential-distributed length with mean 6400 bits).

We can see that DQRAP/CDMA, using 16 spreading codes for all the users, outperforms Slotted-ALOHA/CDMA, using one assigned spreading code per terminal, in terms of the maximum stable throughput and in the delay characteristics. It can manage more than 15% heavier traffic load without entering instability region.

5 CONCLUSIONS

It has been presented a proposal of a near–optimum random access protocol for a CDMA environment suitable for the future third generation mobile communication systems. An analytical model has been introduced and the obtained results fit correctly with those obtained by computer simulations. 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. So it becomes a good proposal for the improvement in the use of the capacities of random access channels in the reverse link.

It has been also shown that the protocol outperforms Slotted-ALOHA/CDMA in terms of the maximum stable throughput and in the delay characteristics, even using a much reduced set of spreading codes. This results are very promising and future research will assess whether this protocol is suitable for other channel conditions.

REFERENCES

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

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

[3] 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.

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

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

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

[7] X. Zhang, G. Campbell, "Performance Analysis of Distributed Queueing Random Access Protocol". DQRAP Research Group Report 93-1. Computer Science Dept. Illinois Institute of Tech. August 1993. http://www.iit.edu/~dqrap/html/papers2.html