On the Efficient Voice-Data Integration over Medium Capacity Wireless TDMA Channels

P.Koutsakis and M. Paterakis
Dept. of Electronics and Computer Engineering
Technical University of Crete
73100 Chania
GREECE

Abstract:- A new medium access control (MAC) protocol for mobile wireless communications is presented and investigated. We explore, via an extensive simulation study, the performance of the protocol when integrating voice and data traffic over a wireless channel of medium capacity (referring mostly to outdoor microcellular environments). Data message arrivals are assumed to occur according to a Poisson process and to vary in length according to a geometric distribution. We evaluate the voice packet dropping probability and access delay, as well as the data packet access and data message transmission delays for various voice and data load conditions. By combining two novel ideas of ours with a useful idea which has been proposed in other MAC schemes, we obtain very good voice sources multiplexing results along with most satisfactory voice and data performance and quality of service (QoS) requirements servicing. This is demonstrated by the nature of our results, as well as by the comparison of both the concepts and the results of our scheme to those of MAC schemes which were recently introduced in the literature.

Key-Words:MAC protocols,Quality of Service,Voice-Data integration,Wireless Channels. Proc.pp..2831-2837

1.Introduction

Future generation wireless personal communication networks (PCN) are expected to provide multimedia capable wireless extensions of fixed ATM/B-ISDN, as data and video traffic will soon gain in importance due to the continuous proliferation of small, portable and inexpensive computing devices.

In this work, we design and evaluate a multiple access scheme that multiplexes voice traffic at the vocal activity (talkspurt) level to efficiently integrate voice and data traffic in an outdoor microcellular environment.

We focus on the uplink (mobiles to base station) channel, where a MAC scheme is required in order to resolve the source terminals contention for channel access.

A well designed multiple access protocol will reduce system costs by maximizing system capacity, integrating different classes of traffic (as opposed to current wireless networks which are mostly optimized for voice communications only), and satisfying the diverse and usually contradictory quality of service (QoS) requirements of each traffic class.

2. System Model

2.1 Channel Frame Structure

The uplink channel time is divided into time frames of equal length. The frame duration is selected such that a voice terminal in talkspurt generates exactly one packet per frame. Each frame consists of three *types* of intervals, the *voice request* interval, the *data request* interval and the *information* interval. Within the information interval, each slot accommodates exactly one, fixed length, packet that contains voice or data information and a header. All request intervals (voice or data) are subdivided into minislots and each minislot accommodates exactly one, fixed length, request packet. For both voice and data traffic, the request must include a source identifier. For data traffic, the request must also include message length in packets and perhaps QoS parameters such as priority.

Since we assume that all of the voice source transitions occur at the frame boundaries, we place the voice request interval at the beginning of the frame, in order to minimize the voice packet access delay. The channel frame model is presented in Figure 1.

The voice and data terminals do not exhaust their attempts for a reservation within the request intervals. Any other free, at the time, information slot of the frame can be temporarily used as an extra request slot (ER slots), with priority given to the voice terminals. This approach is introduced and implemented in [2,7].

By using more than one minislot per request slot, a more efficient usage of the available request bandwidth is possible. *We introduce the idea that certain request slots can be shared by voice and data terminals (first by voice terminals and, after the end of voice contention, by data terminals)*, in order to optimize the use of the request bandwidth.

The concept of reserving a minimum bandwidth for both voice and data terminals to make reservations helps to keep the access delay within relatively low limits and gives clearly better performance than the PRMA [4] and quite a few PRMA-like algorithms (e.g., [6]), where the absence of request slots leads to a continuously decreasing probability of finding available information slots as traffic increases, and hence to greater access delays. A request bandwidth of 2-3% is usually required for optimum system performance.

2.2 Actions of Voice and Data Terminals, and Base Station Scheduling

Voice and data terminals with packets, and no reservation, contend for channel resources using the *two-cell stack* blocked access collision resolution algorithm [9], in order to transmit their request packets only during the voice or data, respectively, request intervals. The base station broadcasts a short binary feedback packet at the end of each minislot indicating only the presence or absence of a collision within the minislot (collision (C) versus non-collision (NC)). It is assumed that the feedback information is immediately available to the terminals (i.e., before the next minislot). Upon successfully transmitting a request packet the terminal waits until the end of the request interval to learn of its reservation slot (or slots). If unsuccessful within the request interval of the current frame, the terminal attempts again in the request interval of the next frame. A terminal with a reservation transmits freely within its reserved slot. Generally, a terminal that fails to transmit a request tries again in successive frames until it succeeds. However, since voice packets that age beyond the voice delay limit are dropped, a voice terminal may stop transmitting requests without ever succeeding.

The base station (BS) allocates channel resources at the end of the request interval, if available. If the resources are unavailable, the request remains queued.

We assume that the BS always allocates the earliest available information slot within the frame, and that it services every outstanding voice request before servicing any data requests. Within each priority class, the queuing discipline is assumed to be FCFS.

Finally, *we apply a low-voice-load mechanism to our scheme*. As data terminals try to transmit messages that vary in length and are, on average, much longer than one packet, it would be both unfair to them and diminishing to our system's performance to not allocate to them more than one slot per frame if resources are available. On the other hand, by allocating more than one slot per frame to data terminals, voice terminals would find a lower number of information slots available for either reservations or requests (ER slots), and our objective is to enforce voice priority. Therefore, we implement the following mechanism.

We define the *frame voice occupancy* as the ratio of {(voice reservations + voice requests) / number of information slots in the frame}. This ratio is calculated by the BS immediately after the end of the voice request slot of each frame. If the ratio is lower than a set limit, we allow data terminals with requests to acquire more than one slot in the current frame. Still, only the first allocated slot is guaranteed to data terminals with reservations in subsequent frames. The selection of the *low frame voice occupancy limit* and of the *maximum number of slots that can be allocated to data terminals within a frame* (the two parameters of our low-load mechanism) must be done carefully, so that even in the case of low voice load enough information slots will still remain available in the next frame for voice terminals who enter talkspurt to use as ER slots.

These selections should be based on the combination of the following two factors:

a) the average data message length, and

b) the channel capacity.

We introduce our numerical choices for the two low-load parameters in Section 3.

2.3 Voice and Data Traffic Models

Voice terminals are equipped with a voice activity detector [3,4]. Voice sources follow an alternating pattern of talkspurts and silence periods (on and off), and the output of the voice activity detector is modeled by a two-state discrete time Markov chain. All of the voice source transitions (e.g., talk to silence) occur at the frame boundaries, and the voice delay limit is equal to the duration of two time frames.

The channel is error-free and without capture, and reserved slots are deallocated immediately by the BS.

Data messages are generated by a large unknown number of data terminals (theoretically infinite). The aggregate message arrivals are Poisson distributed with mean ë messages per frame. The messages vary in length according to a geometric distribution with parameter q and mean $B=1/q$.

3. System Parameters

Our simulations were conducted with the channel and low-load-mechanism parameters contained in Table 1.

4. Results of Voice-Data Integration in a Medium Capacity Channel (VDI-MCC)

To demonstrate the very good performance of our scheme, we will compare it with two previously proposed efficient schemes for voice-data integration, IPRMA [8] and RRA [1,5]. The comparison with IPRMA can only be done conceptually, since the system parameters of the two schemes are completely different, whereas in the comparison with RRA we use the same system parameters.

The IPRMA protocol presents four disadvantages when compared to VDI-MCC. The first disadvantage is the use of the PRMA algorithm to resolve voice terminal contention, as opposed to our use of the two-cell stack algorithm. PRMA is an Aloha-based reservation random access algorithm with constant retransmission probability. As such, it exhibits instability for high loads and achieves lower throughput than the inherently stable tree and stack collision resolution algorithms [10], which are adopted in our system. The second disadvantage is the absence of request slots, the importance of which has been stated earlier[1].

The other two, more important disadvantages of IPRMA are that it does not provide absolute priority to voice traffic over data traffic, and the static nature of its low-load mechanism. The granting, in IPRMA, of a much smaller transmission probability to the data terminals is not enough to ensure absolute voice priority, which however is guaranteed in VDI-MCC. As for the IPRMA's low-load mechanism, the following remarks should be made. In IPRMA, if there are k idle slots in a frame of N slots, the authors impose a speech priority of M slots and a data user who has several packets ready for transmission is allowed to reserve up to $(k\text{-}M\text{-}1)$ slots, thereby keeping a minimum number of idle slots available for speech transmission. This handling of the low voice load situation presents the innate problem of the external parameter M imposition, instead of its dynamic adjustment, which would be best. Imposing the M limitation externally will certainly result, in some frames, in sacrificing slots for use by voice terminals that will not need them. On the contrary, our low-load mechanism is totally dynamic, taking into consideration the voice users needs, the knowledge of which is possible because of the use of the request slots. This way the data users are optimally served, as it is possible for them to acquire, if needed, all the available information slots in the frame.

The almost obligatory absence of request slots in IPRMA does not justify for the lack of a dynamic low-load mechanism. One easily implemented possible approach would be to make an estimation of the number of voice terminals that will try to transmit in each slot of the frame, based on the number of reserved slots in the frame and the probabilities $\mathbf{p}_T$ and $\mathbf{p}_{ST}$ of the voice source model (shown in Figure 2), in a way similar to that of the Controlled Aloha algorithm [1,10]. With the use of such an estimation procedure, data users would be able to dynamically acquire the maximum (or quite close to the maximum) number of available slots of each frame.

The RRA protocol, designed and implemented by A. Cleary et al. [1,5], considered a system model quite similar to ours, with five differences:

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

The comparison of our scheme's results with those of RRA clearly demonstrates our scheme's significantly better performance, as VDI-MCC improves the results on the voice capacity and QoS requirements and remarkably improves the results on the data

[1] However, it should be noted that for the simulation parameters considered in [15] (224 Kbps channel transmission rate, 20 slots per frame) the use of even one request slot would incur a 5% bandwidth overhead (1/20 slots) on the system and would potentially deteriorate, instead of improving, the system performance.

performance and QoS requirements. More specifically, as shown in Figure 2, and as expected by: a) the immediate allocation of resources (after the end of the corresponding request interval, while in RRA allocation is done at the end of each channel frame), b) the use of ER slots, and c) the possible allocation of more than 1 information slots per frame to the data terminals in low voice load situations, the improvements in mean data packet access delays (DaD)[2] and mean data message transmission delays (DmD)[3] over the corresponding delays for RRA are dramatic. Both of these data performance metrics are remarkably lower in VDI-MCC. More specifically, we observe that in RRA the data message delay is consistently greater than the data access delay by about 84 ms. This is because the data message delay in this scheme equals the sum of the access delay and $(B-1) \times F$, where F is the frame duration (i.e., 12 ms). On the contrary, in VDI-MCC the data message delay is greater than the data access delay by just 6.5-10 ms, due to the three basic improvements mentioned above. The minimum difference of 6.5 ms between the two delays in our scheme is explained by the fact that, although data messages whose length is smaller than or up to 8 packets can be transmitted within a few slots (7 at a maximum, beyond the slot in which the first is transmitted), data messages whose length exceeds 8 packets need to wait for more than a frame for the completion of their transmission. As a consequence of these two situations, the average time needed for the transmission of a data message after this message has been allocated its first slot is at least somewhat longer than half a frame (6 ms). Additionally, we see from Figure 3 that the mean data message delay for RRA is maintained below 200 ms until the data message arrival rate ë equals about 2.5 messages/frame, then it increases sharply (about 400 ms for ë=2.55 and off the scale for ë=2.6).

In VDI-MCC, on the contrary, the mean data message delay is impressively lower than that in RRA for ë<2.6, it remains below 200 ms for arrival rates up to 5.76 messages/frame before it starts to increase sharply and eventually goes off the scale for ë=5.82. Consequently, in RRA the maximum data packet throughput achieved with the mean data message delay below 200 ms is about 20 (2.5×8) packets per frame, which corresponds to a 41.7% channel throughput[4] (i.e., 20/48). In VDI-MCC, in turn, the maximum data packet throughput achieved with the mean data message delay below 200 ms is 46.08 (5.76×8) packets per frame, which corresponds to a 94% channel throughput, more than twice as much as the channel throughput of the RRA.

Figure 4 shows the DaD and DmD curves for both schemes, for a constant number of VTs (80), which correspond to a medium and a high voice load, respectively) and for different data message arrival rates. We observe that the mean data message delay, for both medium and high voice loads, is in VDI-MCC not only much smaller than the mean data message delay in RRA, but also smaller than the mean data access delay in RRA (i.e., in VDI-MCC data messages are transmitted faster than the transmission of just one data packet in RRA). Furthermore, for VTs=80 RRA achieves a channel throughput of 90.3% (for ë=1) with the average data message delay below the limit of 200 ms. The corresponding channel throughput result for the VDI-MCC protocol is 91.8% (for ë=1.2). Therefore, the channel throughput in our scheme is consistently greater than that of RRA.

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

Table 2 presents the results for the maximum voice capacity achieved by VDI-MCC for different data message arrival rates, when fulfilling both the QoS requirements of voice and data traffic. It is shown once more that the improvements achieved in our scheme in comparison to RRA, are substantial for all the performance metrics presented in the table .

5. Conclusion

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

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

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

In this paper, a new MAC protocol for mobile wireless communications is presented and investigated. We combine two novel ideas of ours with a useful idea which has been proposed in other MAC schemes, and we achieve very good voice sources multiplexing results along with most satisfactory voice and data performance and quality of service (QoS) requirements servicing. This is shown in detail by the comparison of both the concepts and the results of our scheme to those of MAC schemes which were recently introduced in the literature.

References:

1. A. C. Cleary, ″ Voice-Data Channel Access Integration in Third Generation Microcellular Wireless Networks: Design and Performance Evaluation ″ , *Ph.D. Dissertation, CIS Dept., University of Delaware*, 1997.
2. N. M. Mitrou, Th. D. Orinos and E. N. Protonotarios, ″ A Reservation Multiple Access Protocol for Microcellular Mobile-Communication Systems ″, *IEEE Transactions on Vehicular Technology*, Vol. 39, No. 4, Nov. 1990, pp. 340-351.
3. P. T. Brady, ″ A Technique for Investigating On-Off Patterns of Speech ″, *Bell Sys. Tech. Journal*, Jan. 1965.
4. S. Nanda, D. J. Goodman and U. Timor, ″ Performance of PRMA: A packet voice protocol for cellular systems ″, *IEEE Transactions on Vehicular Technology*, Vol. 40, pp. 584-598, 1991.
5. A. C. Cleary and M. Paterakis, ″ Design and Performance Evaluation of an RRA Scheme for Voice-Data Channel Access in Outdoor Microcellular Wireless Environments ″, *Mobile Networks and Applications (MONET) Journal, ACM and Baltzer Science Publishers*, Vol. 2, No. 1, 1997, pp. 31-43.
6. D. A. Dyson and Z. J. Haas, ″ A Dynamic Packet Reservation Multiple Access Scheme for Wireless ATM ″, *IEEE MILCOM'97*.
7. N. M. Mitrou, G. L. Lyberopoulos and A. D. Panagopoulou, ″ Voice and Data Integration in the Air-Interface of a Microcellular Mobile Communication System ″, *IEEE Transactions on Vehicular Technology*, Vol. 42, No. 1, Feb. 1993, pp. 1-13.
8. W. C. Wong and D. J. Goodman, ″ A packet reservation multiple access protocol for integrated speech and data transmission ″, *IEE Proceedings-1*, Vol. 139, Dec. 1992, pp. 607-612.
9. M. Paterakis and P. Papantoni-Kazakos, ″ A Simple Window Random Access Algorithm with Advantageous Properties ″, *IEEE Transactions on Information Theory*, Vol. IT-35, No. 5, September 1989, pp. 1124-1130.
10. D. Bertsekas and R. Gallager, ″ Data Networks ″, 2nd Ed., Prentice Hall Inc., 1992.

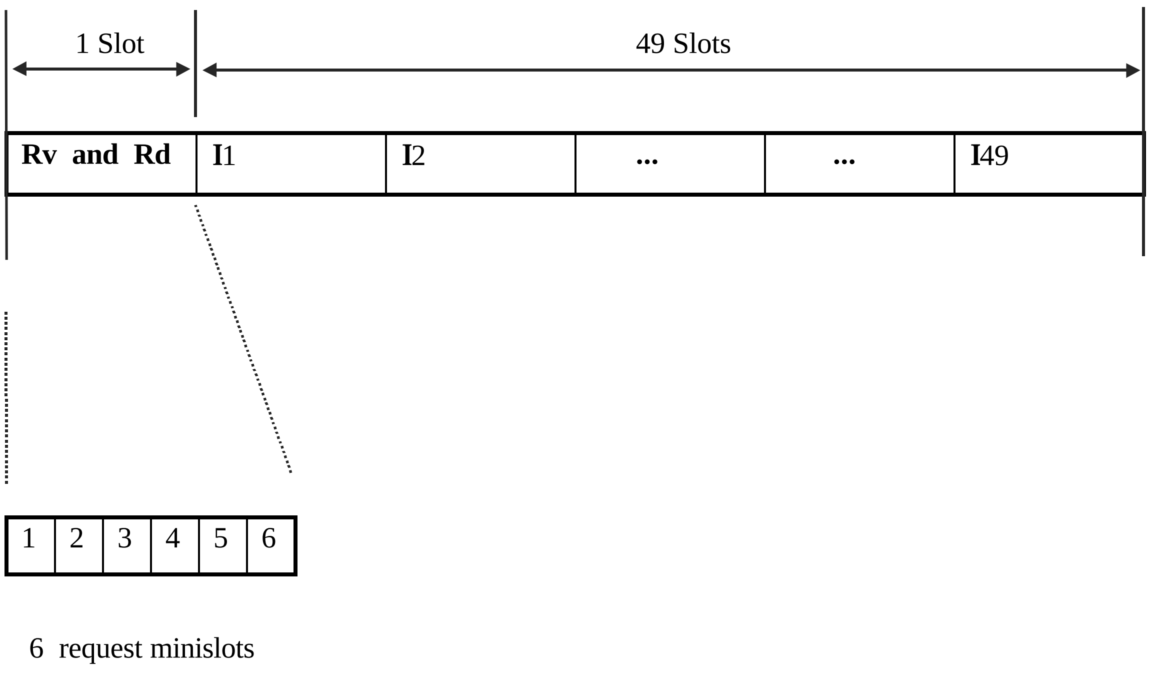

Figure 1. Frame structure for the 1.8 Mbps channel.

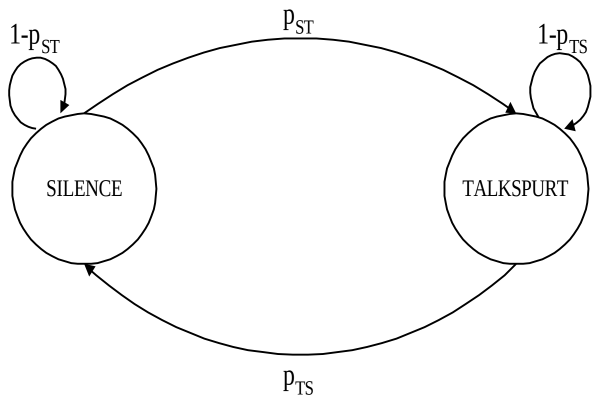

Figure 2. The voice source activity discrete time Markov chain model.

Design Parameters	Medium Capacity Channel
Channel Rate (Mbps)	1.8
Speech Codec Rate (Kbps)	32
Frame Duration (ms)	12
Slots per Frame	50
Slot duration (ì s)	240
Request slots per frame	1
Minislots per request slot	6
Packet size (bytes)	53 (5 header)
Voice delay limit (ms)	24
Mean talkspurt duration (s)	1.41
Mean silence duration (s)	1.78
B (average data message length)	8 packets
Low frame voice occupancy limit	95%
Maximum number of slots allocated to data terminals	8
Max. voice dropping probability	0.01
Max. acceptable average data message delay (ms)	200

Table 1. Experimental System Parameters

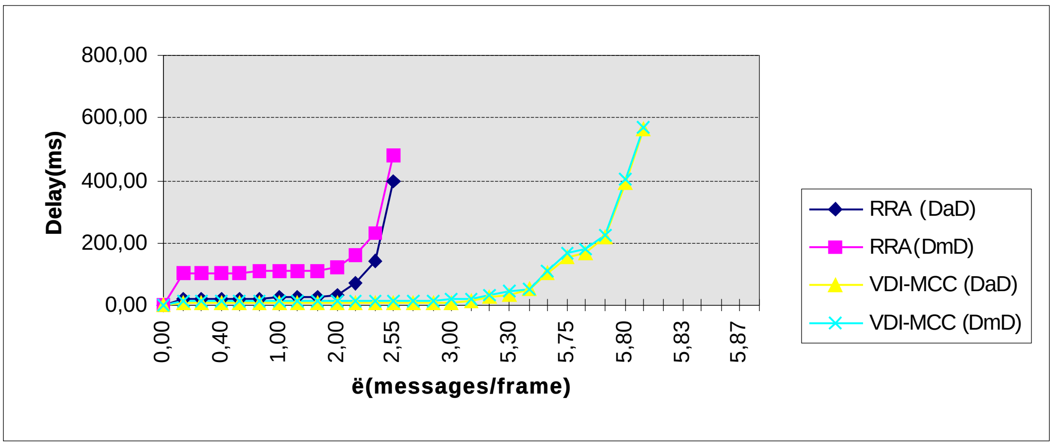

Figure 3. Steady state mean data delays in the absence of voice traffic.

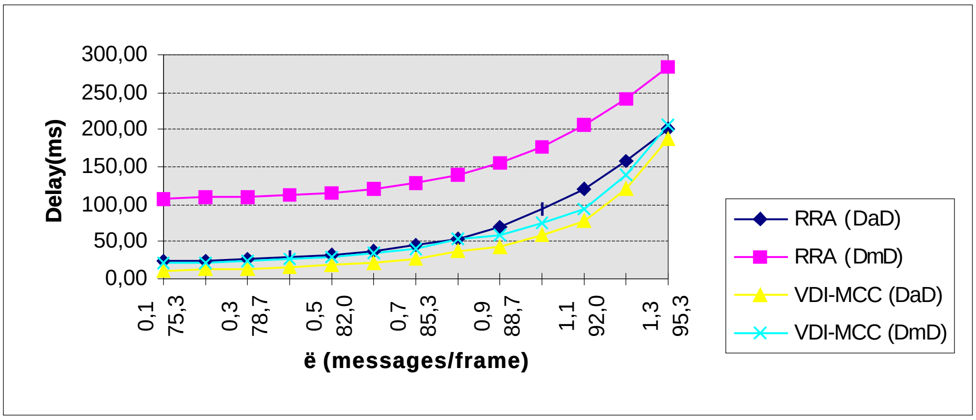

Figure 4. Steady state mean data delays: N=80.

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

Table 2. QoS comparison of VDI-MCC and RRA.