Link Layer Algorithms for Efficient Multicast Service Provisioning in 3G Cellular Systems

Michele Rossi*, Michele Zorzi*, Frank H.P. Fitzek†
*Department of Engineering, University of Ferrara
Via Saragat 1, I-44100 Ferrara, Italy, e-mail: {mrossi,mzorzi}@ing.unife.it
†Department of Communications Technology, Aalborg University
Neils Jernes Vej 12, 9220 Aalborg Øst, Denmark, e-mail: ff@kom.aau.dk

Abstract— **In this work we introduce error control algorithms for multicast delivery in 3G cellular networks. For efficiency reasons, the delivery of multicast flows is usually achieved by allocating a common channel in the forward direction. This enables multiple users to be served by a single physical resource. However, different users are affected by independent channel error processes and new techniques, different from plain ARQ, have to be found to perform error recovery. These new algorithms are needed to deliver the multicast flow in an efficient manner and to enable a reliable, performant and network operator inexpensive multicast service. In the paper different hybrid ARQ algorithms for the error recovery of multicast flows over common channels are proposed and their performance is evaluated both analytically and by simulation. The proposed solutions have been found to be effective and advantageous over plain ARQ techniques. At the end of the paper, results on the achievable video quality are reported for the multicast video streaming case by considering the H.263 video coding format.**

I. Introduction

In radio networks the spectrum is a scarce resource which, in addition, is also limited by the regulation bodies existing in different countries. Hence, it is a pivotal point for network operators to get the highest possible profit out of it. This is especially true for 3G wireless networks, where substantial amounts of money have been invested by network providers to gain access to the spectrum of interest. The dilemma of the network provider is that new services, which would encourage the customers to join the 3G technology, require even more bandwidth. In case the network providers would ask for the same amount of money per bit as they are used to do for voice or paging services, the customers would probably not use it at all. From the operator's point of view, it is therefore very important to offer these new bandwidth demanding services in the most efficient way.

A possible solution to this problem is to enable multicast/broadcast services, where multicast refers to the possibility of using the same spectrum resource for different users. This advantage was already identified by 3GPP [1] by introducing physical channel structures to perform multicast. However, a still unsolved problem is the efficient error recovery for 3G wireless link when these common channel resources are allocated to deliver multicast flows. Taking inspiration from previous work, dealing with multicast error recovery in wired networks [2] [3], in this paper we introduce a link layer algorithm for an efficient multicast service provisioning in 3G cellular systems.

In order to explain and test the proposed scheme, let us assume that a multicast flow, which is originated by a server placed somewhere in the Internet or in the 3G core network, is conveyed towards a set of interested multicast users in the same 3G cell. The serving base station is responsible for the delivery of that flow to all the interested multicast users in its coverage area. Since our main focus here is on the delivery that is taking place on the last hop of the connection, i.e., in the wireless link between serving Base Station (BS) and interested users, we disregard the possible errors occurring over the wired part of the network. We also disregard additional delays and out-of-ordering. Instead, at the BS, we consider an error-free and timely-delivered multicast flow. Now, the problem to be solved is to deliver such a flow over the last hop in an efficient way, i.e., keeping into account both channel efficiency and delay requirements.

The remainder of this paper is organized as follows. In the following Section II the considered reference scenario is introduced and the system simulator, used to derive the channel error processes and test the proposed algorithms, is described. Later on, in Section III two error recovery algorithms for multicast flows in 3G networks are presented. In Section IV, the performance of these algorithms is analytically derived considering an independent error model (iid). The meaning of this section is to characterize these schemes and to emphasize the trade–offs involved in their setting. In Section V some results are reported considering accurate 3G channel traces for the multicast video streaming case. Finally, Section VI concludes the paper.

II. System Under Investigation

We consider here a 3G cellular system, where W-CDMA is used as the radio interface. Moreover, as suggested by [1] we consider that Common CHannels (CCHs) are allocated for the local transmission (at every base station) of multicast data. Obviously, this choice is dictated by the need for an efficient utilization of the channel resources, as motivated above. In order to derive accurate channel traces for this system a W-CDMA cellular system simulator has been developed. The reference scenario together with some details about the simulator are reported in the following.

The service area is composed by $N_c = 9$ hexagonal cells, where a base station is placed at the center of each cell and a given number of users are mobile within the coverage area. Propagation phenomena are modeled through standard techniques, by considering log-normal slow fading, fast fading and path loss [4]. A simple power control algorithm has been implemented following the basic algorithm which can also be found in [5]. In practice, a Signal to Interference Ratio (SIR) target value (SIR_{th}) is assigned to each user in the system. Then the downlink transmission power is dynamically varied by a constant multiplicative increase/decrease factor (Δ).

Considering a specific user, its downlink transmission power is increased when the instantaneous SIR is below SIR_{th}, and it is increased otherwise. Minimum and maximum transmission powers are $P_{min}^{downlink} = -15$ dBW and $P_{max}^{downlink} = -5$ dBW, whereas $\Delta = 0.5$ dB. For what concerns channel coding and interleaving, we consider here a convolutional half rate Viterbi decoder operating over an interleaving interval $TTI = 80$ ms.

A first set of users $N_{DCH} = 200$ is communicating through a Dedicated CHannel (DCH) whose bit rate is 30 Kbps (corresponding to a physical Spreading Factor of $SF_{DCH} = 128$). These users are placed randomly at the beginning of the simulation and are moving following a pseudo-linear mobility model. The power control procedure is dynamically executed for each user as explained above. This first set of DCH users is regarded here as system interference since our main focus is on the common channel multicast transmission.

A second set of users $N_{CCH} = 1000$, receives the multicast flow through a Common downlink CHannel (CCH) whose bitrate and spreading factor are indicated here as $B_r = 120$ Kbps and $SF_{CCH} = 16$, respectively. These users can also be on the move, but their serving base stations remain unchanged. Let us better clarify this point. CCH users are randomly placed at the beginning of the simulation, shadowing and path loss are chosen according to the log-normal and the exponential model, respectively, and are kept constant for the whole simulation time. Subsequently, in order to emulate some degree of mobility, their Doppler frequency f_d is selected, but without changing their spatial coordinates. In this way, we are able to control their fast fading as if they were on the move but without reflecting it into a change of their spatial positions. Therefore, it is possible to investigate multicast delivery algorithms (which is, in fact, the main focus here) disregarding the multicast handover management that, by itself, constitutes a problem to be properly handled. The common channel power has been fixed to the constant value $P_{CCH} = P_{max}^{downlink}$ throughout the whole simulation.

III. Link Layer Algorithms for Multicast Streaming

When the multicast flow is transmitted by means of a common channel, different users are in general characterized by independent channel error processes. This is very important since it can make traditional Link Layer (LL) retransmission approaches inefficient as the multicast group size (N_u, in the system under investigation $N_u = N_{CCH}/N_c$) increases.

For illustration purpose, a fully reliable multicast service is considered at this point. In that case, a packet needs to be retransmitted if at least one user has not correctly received it. Moreover, an erroneous packet needs to be retransmitted until all the users in the multicast group have received it correctly. The problem is that, as N_u increases, the probability that at least one user needs a retransmission at a given time increases as well and, as a consequence, the forward link throughput is heavily degraded by retransmissions, while the available bandwidth for new transmissions becomes very low.

As N_u increases, simple ARQ (Automatic Retransmission reQuest) may not be sufficient to cope with this problem, because it simply retransmits the lost packets whenever a retransmission request (Not Acknowledgment, NACK message) arrives at the base station, but it does not account for the possibly different error processes that, at every receiver, affect the transmitted packet. To cope with this problem, we propose to exploit packet-based FEC techniques directly at the link layer. A rich literature can be found on the topic [2] [3] [6] [7].

By this approach, some error recovery is performed directly at the receiver side. This is realized by pro-actively adding some redundancy into the multicast flow. This redundancy can be independently exploited, at each receiver, to recover from losses. If local recovery succeeds, no retransmissions are required and the forward bandwidth can be utilized to allocate new packet transmissions. In more detail, the multicast flow at the link layer of the sending base station is divided in transmission groups (TG) of K packets each. Then, each group is passed to a packet–based encoder where H redundancy packets are generated for every TG. Finally, the whole FEC block composed by $K+H$ packets is sent over the channel. At the receiver side, every user can exploit the redundancy packets by recovering up to H erroneous or lost packets, in any order. Different coding schemes can be used for this purpose, Reed–Solomon and Tornado deserve a particular attention [8], [9]. However, in this work we are mainly interested in discussing the effectiveness of these coding strategies without going into code implementation details. Moreover, in the following, we assume that, in the first encoding phase, it is possible to generate a large (but finite) number of redundancy packets for each FEC block. These packets will be transmitted on-demand as incremental redundancy during the error recovery algorithm. The main benefits of this approach are:

- *Improved transmission efficiency*: A single parity packet can be used to repair the loss of any packet in the TG. This means that *a single parity packet* can repair the loss *of different data packets at different receivers*. This fact is extremely useful since different receivers are in general affected by independent error processes.
- *Improved scalability in terms of group size*: In ARQ schemes the sender needs to know the sequence number of each lost packet. Instead, using parity packets for loss repair, the sender only needs to know the maximum number of lost packets by any receiver but not their sequence number. So, the feedback is reduced from per–packet feedback to per–TG feedback. In fact, depending on the number of lost packets, a given number of new redundancy packets, obtained from the original K packets, can be transmitted over the channel (incremental redundancy) so that the original data is recovered if at least K packets are correctly received among all the received packets for the TG (N FEC block packets plus incremental redundancy packets).
- *Improved feedback channel performance*: Thanks to the pro-actively added redundancy, some error recovery is possible at the user terminals, without the need for retransmissions. As a consequence, the number of acknowledgment messages that the users are sending back to the base station is considerably decreased. This FEC beneficial effect has been largely investigated in previous work (see [10] as an example), where techniques to limit the ACK collision problem have been considered. These results and algorithms still apply in our scenario as well. Results on ACK collision avoidance will not be explicitly considered in our contribution where the emphasis is mainly put on the proposal and the presentation of schemes for the multicast

delivery in 3G networks. Performance investigation is then given considering forward channel metrics.

The hybrid ARQ (HARQ) algorithms considered throughout this paper are presented in the following:

HARQ1: At the sender side TGs of N PDUs each (K data packets plus H redundancy PDUs) are sent first. Then, each receiver checks for errors in each TG and replies accordingly. If a receiver detects less than K correct PDUs for a TG it sends back to the sender a NACK including the TG identifier. The sender collects incoming NACKs and, if the number of collected NACKs for a given TG is greater than zero the following procedure is activated:

- The K original PDUs included in the erroneously received TG are fed again to the packet–based encoder to obtain $\xi \geq 1$ redundancy packets, i.e., ξ redundancy packets for that TG that are however different from all redundancy packets previously transmitted.
- The ξ new redundancy PDUs (*incremental redundancy*) are sent over the CCH channel.

In practice, we have that the number of retransmitted redundancy packets for each received NACK is always equal to ξ. In this scheme, each receiver collects all the received packets for a TG, i.e., the N PDUs sent in the first TG transmission plus the, say R, redundancy PDUs sent in the following retransmissions (triggered by NACKs). The original K PDUs in a TG can be recovered if the number of correct PDUs, N_{ok}, over the $N+R$ PDUs is greater than or equal to K. This scheme, when ξ is set to one, tries to maximize the channel efficiency. In fact, at each retransmission request the minimum amount of redundancy (one packet) is retransmitted, thereby limiting as much as possible the probability to retransmit useless redundancy packets. However, if at least one user needs more than one new PDU to obtain the K original data packets he will require a further retransmission. For this reason this scheme is also characterized by a large delay for the correct delivery of a TG.

HARQ2: At the sender side the TG of N PDUs (K data packets plus H redundancy PDUs) is sent first. Each receiver checks for errors in each TG and replies accordingly. To better explain how the algorithm works suppose that, in addition to the N PDUs in the first transmission, R redundancy PDUs have already been sent over the channel for a given TG. In this case, each receiver checks for the number of correctly received PDUs (N_{ok}) among the $N+R$ PDUs sent. Let us refer to a given user $i \in \mathcal{N}$, where $\mathcal{N}$ is the set of multicast users in the cell. If $N_{ok}(i) \geq K$ the original K PDUs can be obtained and the TG is correctly received by user i. Otherwise, if $N_{ok}(i) < K$, user i sends back a NACK including the *TG identifier* and $r_i = K - N_{ok}(i)$, i.e., the minimum number of new redundancy PDUs needed for the correct decoding of the K data PDUs at that terminal. The sender collects incoming NACKs and computes $R_{max} = \max_{i \in \mathcal{N}}(r_i)$. Then, $R = R_{max}$ new redundancy PDUs are encoded for that TG and are transmitted over the CCH channel. This procedure is repeated until all users in $\mathcal{N}$ are able to correctly decode the K data PDUs, i.e., when $N_{ok}(i) \geq K \ \forall i \in \mathcal{N}$. In this algorithm the minimum number (channel efficiency $\uparrow$) of PDUs needed to ensure that all multicast receivers will be able to recover the original K packets is sent (delay $\downarrow$). This is performed with the aim of getting a good trade–off between throughput and delay. Note that R_{max} new PDUs guarantee the successful decoding of a TG by all receivers only if no channel errors occur, or if channel errors are such that each receiver is able to decode at least r_i PDUs out of the R_{max} transmitted.

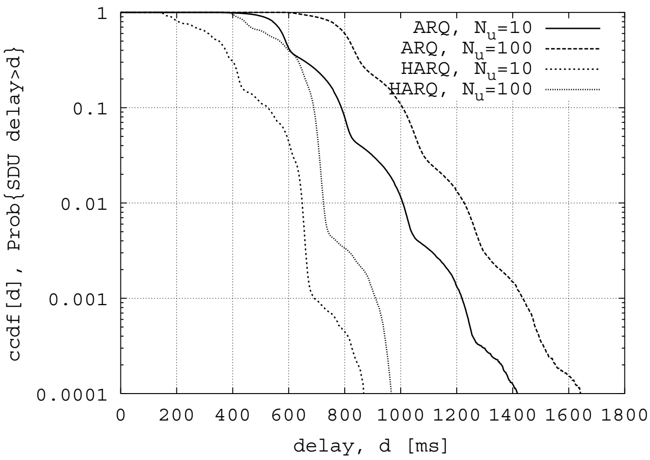

Fig. 1. SDU complementary cumulative delivery delay statistics (ccdf) by varying N_u, $p = 0.1$, independent PDU error processes. Comparison between HARQ3 FEC(K=15, N=19) scheme and selective repeat ARQ.

IV. Performance of Error Recovery Algorithms over an Independent Channel

In this section, we report some preliminary results for the algorithms introduced above. A simple independent channel model will be considered in this Section, due to the possibility to obtain analytical expressions from which useful insights can be derived. Accurate channel traces, obtained from the simulation tool explained in Section II, will be considered later on in Section V to test the HARQ schemes over accurate W-CDMA channel traces.

The performance metrics that we are looking at in this Section are: the *channel efficiency* (CCH channel throughput) and the *higher layer packet delivery delay*. The higher layer packet is the LL SDU unit, that is the packet passed to the LL level to be processed and sent over the channel by higher protocol levels. With the term SDU delay we refer to the time elapsed between the transmission of the first LL PDU composing one SDU to the instant in which the full SDU has been correctly received by every user in the cell.

In the results discussed in the following Sections, we consider a LL logical bit rate of $B_r = 120$ Kbps ($SF_{CCH} = 16$) a LL round trip time (RTT) of 220 ms (this the maximum value for the LL RTT in a 3G network and it is due to the large interleaving depth of $TTI = 80$ ms) and a LL PDU length of 360 bits. With these values, we have that about 77 PDUs are transmitted in a LL RTT. We consider a fixed LL SDU packet length of 500 bytes which is a typical value for the mean frame length of video streaming flows [11]. Moreover, with the term ARQ, we refer here to the standard Selective Repeat ARQ algorithm, where a retransmission for a packet is scheduled if at least one user in the multicast set is requiring for its retransmission.

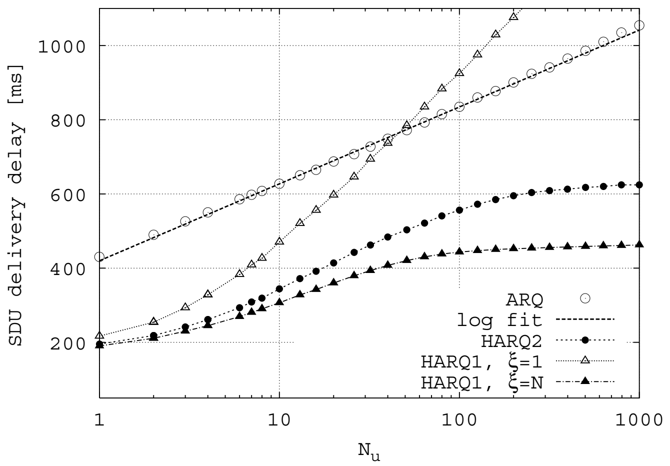

Fig. 2. Mean SDU delivery delay as a function of N_u, $p = 0.1$, *iid* channel. Comparison between HARQ FEC(K=15, N=19) schemes and selective repeat ARQ.

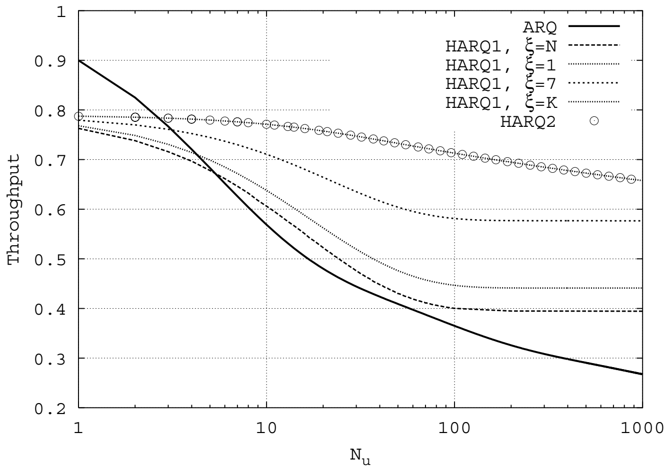

Fig. 3. Throughput as a function of the number of multicast users in the cell N_u, independent channel, $p = 0.1$, FEC(K=15, N=19).

As a first result, in Figure 1 we report the SDU complementary cumulative delivery delay distribution (ccdf) for the HARQ2 and a standard ARQ algorithms. The number of multicast users is set to $\{10, 100\}$ and the PDU error probability is considered to be equal to $p = 0.1$ for every user. From this figure we can observe that as the group size increases (N_u), hybrid ARQ largely limits the SDU delays. In particular, the SDU delivery statistics, after a certain point (in $d \leq 800$ ms), starts decreasing very suddenly. The statistics regarding the simple ARQ, instead, is shifted to the right without any shape change. This means that hybrid ARQ techniques are more robust with respect to an increasing number of multicast users in the system.

In Figure 2, we report the mean SDU delivery delay, defined as the mean time needed to correctly transmit a full SDU to all users in the multicast group as a function of the multicast group size, N_u, considering that all users are characterized by an independent channel with the same PDU error probability $p = 0.1$. From this figure it can be observed that in the simple ARQ case, the behavior of the mean SDU transmission time is logarithmic in N_u (approximately equal to $420 + 90\ln(N_u)$). In other words, the SDU delivery delay tends to infinity as N_u increases. Also in the hybrid ARQ case the SDU delivery delay tends to infinity as $N_u \to \infty$, but here the delay increases very slowly with N_u. This is an important fact since it considerably improves the system scalability in terms of multicast group size. As expected, the HARQ1 scheme with $\xi = 1$ is the one with the lowest delay, HARQ1 ($\xi = N$) is the one with the longest and HARQ3 is a compromise solution.

As the last result in this section, we focus on the channel efficiency η, which is defined as the number of PDUs correctly transmitted over the common channel divided by the total number of transmitted PDUs:

$$\eta = \lim_{t \to +\infty} \frac{N_{ok}(t)}{N_{tot}(t)} \tag{1}$$

where $N_{ok}(t)$ and $N_{tot}(t)$ are the number of correctly received and the total number of transmitted PDUs in the interval $[0, t]$, respectively. In $N_{ok}(t)$, each packet is counted only once, even if multiple copies could have been sent during retransmissions. Redundancy packets are not accounted for in N_{ok}, whereas they are counted in N_{tot}. In the following, we present a simple way to analytically derive the channel efficiency under the independent channel error assumption. We label as $i \geq 0$ the transmission round for a FEC block, where $i = 0$ correspond to the first transmission, whereas $i \geq 1$ is used to track the following retransmission rounds. Moreover, we consider that, at every retransmission, a constant number ξ of redundancy PDUs is encoded for the TG and sent over the channel. Note that, the efficiency of the HARQ2 algorithm is equal to the efficiency of HARQ1 when $\xi = 1$ since, in both schemes, the minimum amount of redundancy is sent to accomplish error recovery. The only difference among these two algorithms is that retransmissions are distributed differently. However, since the channel is independent, the retransmitted PDU's position is irrelevant, while their number is the only quantity affecting η. Now, we introduce some quantities that will be used in the analysis: N_u is the number of multicast users, i.e., the number of users served by the common channel and p_i is the PDU error probability for user $i \in \{1, 2, \ldots, N_u\}$. The probability that a TG is correctly received by every user in a transmission round up to and including round $j \geq 0$ can being computed as:

$$P[\leq j] = \begin{cases} \prod_{i=1}^{N_u} (1 - p_i^{j+1}) & \text{ARQ} \\ \prod_{i=1}^{N_u} (1 - f[p_i, j]) & \text{HARQ} \end{cases} \tag{2}$$

where:

$$f[p, j] = \sum_{e=N+\xi j-K+1}^{N+\xi j} \binom{N + \xi j}{e} p^e (1-p)^{N+\xi j - e} \tag{3}$$

where the function $f[\cdot, \cdot]$ is used to compute the probability that less than K PDUs have been correctly received among the $N + \xi j$ PDUs transmitted for the TG up to and including round j. This quantity corresponds to the probability that additional redundancy PDUs are still needed to correctly decode the

TG after j transmission rounds, i.e., that at least a further retransmission round is needed. The average number of retransmissions is given by:

$$E[\text{retx}] = \xi \sum_{j=0}^{+\infty} (1 - P[\leq j]) \quad (4)$$

where $\xi = 1$ for the ARQ scheme, whereas $\xi \geq 1$ for HARQ. Finally, the channel efficiency can be derived as:

$$\eta = \begin{cases} \dfrac{1}{1 + E[\text{retx}]} & \text{ARQ} \\ \dfrac{K}{N + E[\text{retx}]} & \text{HARQ} \end{cases} \quad (5)$$

Figure 3 gives the channel efficiency as a function of N_u considering $p_i = 0.1$, $\forall i \in \{1, 2, \ldots, N_u\}$. In the figure, we report the throughput for the HARQ1 scheme proposed above for several ξ values. As observed above, the HARQ2 (that in the iid case is equal to HARQ1 with $\xi = 1$) scheme is characterized by the highest throughput. Moreover, it is worth observing that the reduction of the delay in the HARQ1 scheme with $\xi = N$ comes at the cost of sending a large amount of incremental–redundancy packets. This increases the probability of recovering a TG in a short time, but at the same time decreases the throughput, since more unnecessary packets are sent at each retransmission request. All the hybrid algorithms outperform simple ARQ as N_u becomes greater than three. On the other side, as the multicast group is smaller than $N_u = 3$, the pro–actively added redundancy in HARQ1 and HARQ2 (*a priori* data protection) is more channel consuming than performing retransmissions only (simple ARQ). However, it is interesting to observe that hybrid ARQ schemes lead to a higher throughput as N_u increases and that, for large N_u values, they still achieve an acceptably high channel efficiency, whereas simple ARQ in such a case has very poor throughput performance. Another interesting result is that the performance of scheme HARQ2 lies on the throughput upper bound (HARQ1 with $\xi = 1$). This scheme may be a good candidate to be effectively used for multicast data delivery since it is characterized by a good channel efficiency and also its delay performance is not too much worse with respect to the HARQ1 algorithm with large ξ values.

V. VIDEO SERVICES

In this section we report some results on the effectiveness of the proposed algorithms in case H.263 [12] video streams are transmitted over the forward common channel. The results will be expressed here using the picture signal to noise ratio (PSNR) [11] as the performance indicator for the received video quality. For PSNR calculation we compare the encoded/decoded video stream with the encoded/conveyed/decoded video stream.

A PSNR of 100 dB is used to represent the perfect quality case, i.e., when the original and the transmitted streams do not differ. The focus will be mainly on the advantages of the proposed HARQ solutions with respect to classical ARQ algorithms. The following system parameters will be considered in our analysis: number of DCH users in the system $N_{DCH} = 200$, number of multicast (CCH) users per cell $N_u = 50$, High mobility scenario (multicast users Doppler frequency $f_d = 40$ Hz). The video sequences considered in this work to carry out the PSNR performance evaluation is the *Highway* QCIF sequence [13].

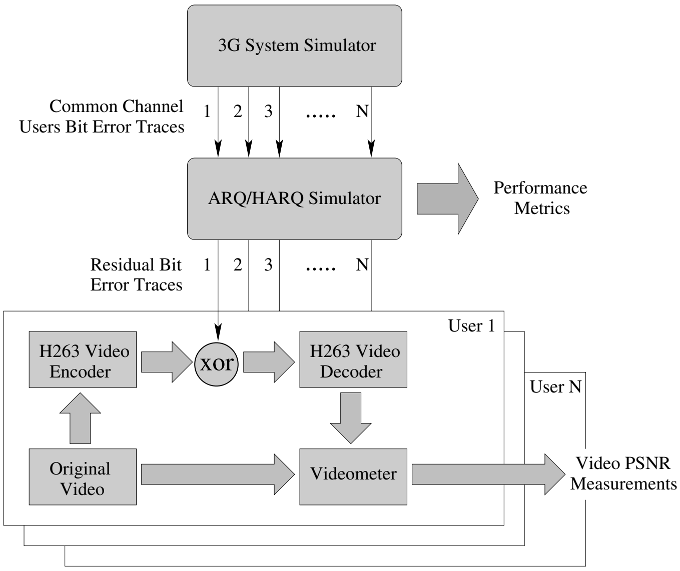

Fig. 4. Methodology considered to carry out Video PSNR measurements.

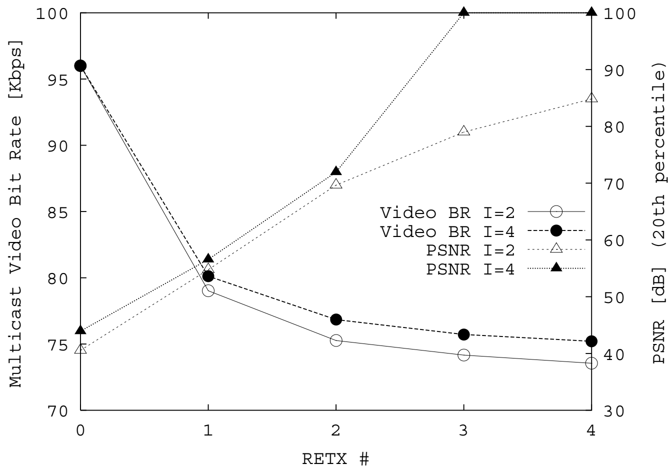

Fig. 5. Video PSNR and useful bandwidth as a function of the maximum number of allowed retransmission rounds per FEC block. HARQ settings: $K = 32$, $H = 8$, $I = 2$ FEC blocks.

In the results reported in the following inter–block interleaving has been considered. In order to increase the FEC robustness over bursty channels, I FEC blocks are interleaved together by mean of a matrix interleaver approach. As a first set of results, in Fig. 5 we report the PSNR and the useful bandwidth as a function of the maximum number of allowed retransmission rounds in the error control algorithm. With the term useful bandwidth, we mean the portion of the downlink channel bandwidth that, on average, can be exploited for the transmission of new data (the maximum bandwidth is $B_r = 120$ Kbps). In Fig. 6 we focus on the 50-th percentile of the users by plotting the following metrics: histograms are used to report the mean PSNR value; on top of every histogram bar, vertical lines are used to report the standard deviation. In

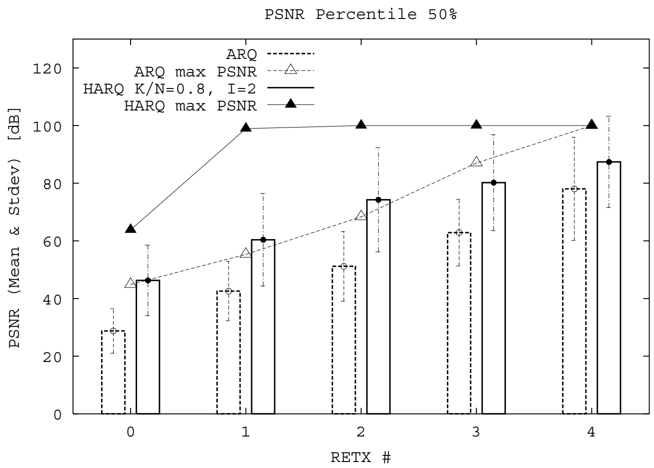

Fig. 6. Mean and maximum PSNR for the 50-th percentile of the users as a function of the maximum number of allowed retransmissions. HARQ settings: $K = 32$, $H = 8$, $I = 2$ FEC blocks.

addition, horizontal lines are drawn to indicate maximum PSNR values for the set of users under analysis.

To obtain the PSNR performance, the original YUV video sequence has been encoded in the H.263 format. Then the residual bit error traces, obtained at the output of the HARQ simulator, have been used to corrupt the H.263 video sequence. Finally, the corrupted video sequence has been decoded into the YUV format again and compared with a decoded version without any errors (see Fig. 4). By not using the original YUV sequences for comparison, we separate the impact of the encoder settings and the impact of the wireless channel. A video-meter tool [14] has been used to compute the differences between the two videos and to translate them into PSNR values. In case of an error free transmission the video quality is set to 100 dB by the videometer tool. This procedure has been repeated for every multicast user in the system. From Figs. 5 and 6, the following observation can be made:

The PSNR performance quickly saturates, i.e., two retransmission rounds seem to be enough to achieve satisfactory PSNR values (PSNR $\geq$ 60 dB, Fig. 5). The inter-block interleaving is effective, especially after the second retransmission round. Even if not reported here, also the interleaving has a negative effect on the delay performance (and play–out buffer dimensioning). In accordance with the PSNR saturation, also the useful bandwidth becomes flat after the third retransmission round. This indicates that further retransmissions are rarely required and that, for this reason, are leading to a small performance increase. An increase of the K/N ratio is beneficial for the useful bandwidth (a gain of almost 10 Kbps is observed as K/N increases from 0.8 to 0.9), but is highly impacts the delay performance (and the play-out buffer occupancy, these results are not reported here due to space constraints).

Finally, from Fig 6 we can observe that the mean PSNR value for HARQ often dominates the maximum PSNR values of plain ARQ. In general, with HARQ solutions two retransmissions are enough to obtain an acceptable video quality (PSNR $\geq$ 60 dB) even for low percentiles (i.e., 20-th percentile not reported here due to space constraints), whereas such performance is possible only after three or four retransmission with ARQ. This implies that the number of retransmission rounds can be decreased when HARQ is being used, by conserving a satisfying performance level and, at the same time, by decreasing the link layer delays (i.e., decreasing the requirements for the receiver play–out buffer). As an example, observe that one retransmission is enough, in the HARQ case, to achieve a perfect maximum quality (PSNR=100 dB) for 50 % of the multicast users in the system, but four retransmissions are needed by the ARQ scheme, i.e., at least one half of the users will be affected by errors up to the fourth retransmissions.

VI. Conclusions

In this paper the multicast delivery in 3G Cellular Networks has been investigated in detail. By means of analysis and simulations we have shown that the proposed link layer hybrid ARQ scheme outperforms standard ARQ techniques. Later on we focused on the multicast video transmission case showing that, even in this case, consistent performance improvements are possible, i.e., the same services can being offered with reduced delays and higher channel efficiencies. The results presented in this paper are only a first step towards the definition of algorithms to be effectively used in future systems. Further aspects, that need to be studied in detail, are related to the feedback channel optimization and to the possibility of offering unreliable services while controlling the minimum video quality at every receiver. These aspects are currently under study.

References

[1] "3GPP Third Generation Partnership Project." [Online]. Available: http://www.3gpp.org

[2] C. Huitema, "The case for packet level FEC," in *International Workshop on Protocols for High Speed Networks (PsHSN 1996)*, INRIA, Sophia Antipolis, France, October 1996, pp. 109–120.

[3] J. Nonnenmacher, E. Biersack, and D. Towsley, "Parity-based loss recovery for reliable multicast transmission," *IEEE/ACM Trans. Networking*, vol. 6, no. 4, pp. 349–361, Aug. 1998.

[4] M. Zorzi, M. Rossi, and G. Mazzini, "Throughput and energy performance of TCP on a wideband CDMA air interface," *Wiley Wireless Commun. and Mobile Computing (WCMC)*, vol. 2, no. 1, pp. 71–84, Feb. 2002.

[5] H. Holma and A. Toskala, *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*. John Wiley & Sons, Ldt, 2000.

[6] A. J. McAuley, "Reliable Broadband Communications Using a Burst Erasure Correcting Code," in *Proceedings of ACM SIGCOMM 1990*, Philadelphia, PA, US, Sept. 1990, pp. 287–306.

[7] J. Nonnemacher, "Reliable Multicast Transport to Large Groups," Ph.D Thesis, Ecole Polytechnique Federale de Lausanne, France, 1998.

[8] L. Rizzo, "Effective Erasure Codes for Reliable Computer Communications Protocols," *ACM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, Apr. 1997.

[9] J. W. Byers, M. Luby, and M. Mitzenmacher, "Accessing multiple mirror sites in parallel: Using Tornado Codes to speed up downloads," in *Proceedings of IEEE INFOCOM*, New York, US, Mar. 1999, pp. 275–283.

[10] D. D. Lucia and K. Obraczka, "Multicast feedback suppression using representatives," in *Proceedings of IEEE INFOCOM*, Kobe, Japan, Sept. 1997, pp. 463–470.

[11] F. Fitzek, P. Seeling, M. Reisslein, M. Rossi, and M. Zorzi, "Investigation of the GoP Structure for H.26L Video Streams," Technical Report, University of Ferrara, Italy, Dec. 2002. [Online]. Available: http://www.eas.asu.edu/˜mre/GoP.pdf

[12] Signal Processing & Multimedia Group, The University of British Columbia, "H.263/H.263+ Research Library, Release 0.2." [Online]. Available: http://spmg.ece.ubc.ca

[13] F. Fitzek and M. Reisslein, "MPEG-4 and H.263 Video Traces for Network Performance Evaluation," *IEEE Network*, vol. 15, no. 6, pp. 40–54, Nov./Dec. 2001. [Online]. Available: http://http://trace.eas.asu.edu

[14] F. Fitzek, P. Seeling, and M. Reisslein, "VideoMeter Tool for YUV bitstreams," *IEEE Network Mag., software tool for networking*, vol. 17, no. 1, Jan. 2003. [Online]. Available: http://www.eas.asu.edu/˜mre/