

Congestion Control and Adaptive Retransmission for Multimedia Streaming over Wireless Networks

Kamal Deep Singh^{*}, Árpád Huszák[†], David Ros[§], César Viho^{*} and Gábor Jeney[†]
^{*} IRISA/INRIA, Rennes, France [†] BUTE, Budapest, Hungary [§] Telecom Bretagne, France
{ksingh, viho}@irisa.fr {huszak, jeneyg}@hit.bme.hu dros@telecom-bretagne.eu

Abstract

In wireless networks, congestion control, alone, may not be enough to ensure good quality of multimedia streaming and efficient utilization of the network. Packet losses due to the high bit error rate not only degrade the multimedia quality, but render the current congestion control algorithms as inefficient: these algorithms back-off on every packet loss even when there is no congestion.

In this paper we study the case of pre-recorded multimedia streaming over wireless networks. We integrate the congestion control schemes with an adaptive retransmission scheme in order to selectively retransmit some lost multimedia packets. Moreover, we integrate a wireless loss estimation scheme to improve the efficiency of congestion control. Our results show that the integrated scheme improves the multimedia quality due to the retransmission and recovery of some of the lost multimedia data. Moreover, the congestion control schemes, and hence the integrated retransmission scheme, show improved performance because of the wireless loss estimation.

Index Terms—congestion control, multimedia streaming, retransmission, wireless loss estimation

1. Introduction

In wireless environment, the frequent losses degrade the quality of the video streams, therefore, new methods should be investigated to increase the number of successfully received packets. The end-to-end packet losses should either be prevented or subsequently handled. Multimedia applications may use congestion control in order to avoid congestion collapse of the network and to minimize the packet loss due to congestion. Nevertheless, the performance of congestion control protocols may significantly degrade over wireless links because they confuse the wireless

losses with the congestion losses and unnecessarily reduce the throughput.

Traditional error control schemes generally use retransmission to provide reliability at the expense of latency. Loss tolerant multimedia applications should use retransmissions as well, but it will be successful only if the retransmitted packet is received before its playout deadline. In a congested network the round-trip-time (RTT) and the loss probability is significantly higher and the retransmitted packets probably will be lost again, therefore the retransmission of lost video stream packets is not recommended in such conditions. To efficiently control the retransmissions a selective retransmission scheme is needed.

Some prior work has been done to develop error recovery and concealment for real-time video. Content based retransmission methods retransmit only the important data of the bitstream, taking advantage of the motion prediction loop employed in most motion compensation based codecs. Correcting errors in a reference frame caused by earlier packet loss, prevents error propagation. In [1] a content based approach is analyzed using SR-RTP [2]. A new selective retransmission scheme for multimedia transmission is proposed [3] over noisy wireless channel using the ATM ABR service. Attempts were made to implement a selective retransmission protocol with a decision algorithm [4] based on the Euclidean distance calculated by the loss and latency ratio. Moreover, the retransmission requests are used in the scheme introduced in [5]. A survey and taxonomy of retransmission-based partially reliable transport protocols can be found in [6].

We propose a combination of the retransmission approach with new congestion control methods that can distinguish between the congestion and the wireless losses. Depending on the calculated sending rate, the video bitrate and the MPEG frame type, we decide whether to enable or disable the retransmissions. In our proposal, we use DCCP (Datagram Congestion Control Protocol) [7], [8] as the

transport protocol because of its benefits that are explained in the following Section 2. In Section 3, we investigate our congestion controlled selective retransmission method for loss tolerant multimedia applications. The obtained results are presented in Section 4. Finally, Section 5 summarizes and concludes this paper.

2. Congestion Control Algorithms

Multimedia applications should use some form of congestion control, both in wired and cellular networks, in order to adapt the sending rate to the available bandwidth. Today's Internet stability is due to TCP and its congestion control algorithm. TCP represents a very efficient transport protocol in general and is suitable for data transfer. However, it has been argued [9] that TCP is unsuitable for video streaming because strict delay and jitter requirements of video streaming are not respected by TCP. Moreover, some TCP retransmissions are unnecessary for video when data may miss the arrival deadline and become obsolete. This has led researchers to look for alternative options. Most of the work related to congestion control for video flows has either emulated TCP or has used the TCP model.

The well-known TCP-Friendly Rate Control (TFRC) congestion control consists in an equation-based rate control mechanism [9][10][11], designed to keep a relatively steady sending rate while still being responsive to congestion. When used over wireless links, TFRC and TCP cannot distinguish between the wireless losses and the congestion losses. They both may suffer from the link underutilization if the connection traverses a wireless link. This is because they consider dropped packets as a sure sign of congestion and reduce the sending rate significantly. The inability to identify a wireless loss followed by unnecessary reduction in sending rate results in link underutilization.

2.1. Analytical Rate Control (ARC)

To the best of our knowledge, ARC [12] is the first to model the ideal behaviour of TCP facing wireless losses. ARC is a rate-control scheme that uses the following equation:

$$S = \frac{1}{4RTT} \left(3 + \sqrt{25 + \frac{24}{p_c}} \right) \quad (1)$$

where S is the sending rate in packets per second, RTT is the round trip time, and p_c is the congestion loss probability. The latter is related to the total packet loss

probability π and the wireless loss probability ω through the expression:

$$p_c = \left(\frac{\pi - \omega}{1 - \omega} \right) \quad (2)$$

ARC needs a way to calculate π and ω , in order to further compute p_c ; note that the value of π is easily estimated from the total packets received and the total packets lost, which in turn can be known by the receiver by looking at the sequence numbers. ARC relies on the MAC layer to calculate ω . However, this approach violates the end-to-end paradigm and will not work if there is no way to obtain the wireless loss probability from lower layers. This disadvantage leads us to the approach discussed in Section 2.2.

We implemented ARC over DCCP, explained in Section 2.2, and TFRC over DCCP for the comparison tests in our simulations. For ARC, some mechanisms from TFRC, like "slow start" emulation, are also integrated. One important point when using ARC is that in our simulations we assume that ARC has the perfect knowledge of ω and just needs to calculate π in order to estimate the value of p_c . Moreover, in the case of ARC, another crucial point is that, unlike TFRC that stops the "slow start" as soon as it detects the first loss, here first loss could be due to wireless losses. Thus, a separate slow start history is maintained and slow start is stopped when $loss\ rate > \omega$ and not when > 0 .

2.2. Wireless Loss Estimation in DiffServ integrated (WLED) with ARC

WLED [13] is an end-to-end wireless loss estimation in DiffServ networks supporting Assured Forwarding (AF) based services. The scheme is designed to help congestion control schemes over wireless networks and assumes the "presence of DiffServ-aware" streaming. The idea of WLED is to exploit additional information regarding the character of losses. In "DiffServ-aware" video streaming the applications "mark" the packets as *green*, *yellow* and *red*. The important packets are marked as *green* and the least important packets as *red*. During congestion, the "staggered Red In and Out (RIO)" [14] Active Queue Management (AQM), used to implement AF Per Hop Behavior (PHB), protects the *green* packets at the cost of dropping *red* and *yellow* packets. The protection of higher-priority packets inherent in the "staggered RIO" algorithm means that, if the loss rate of low-priority packets is not significant, then we may assume that the loss of high-priority packets, *green*, is highly correlated with the wireless loss rate. This idea can be used to estimate the wireless loss and further to obtain the value of p_c in the equations (1) and (2), in

end-2-end manner. Thus, we use WLED in conjunction with ARC and call it WLED-ARC.

Moreover, we use Datagram Congestion Control Protocol (DCCP) [7] as the transport layer protocol for testing the congestion control and the retransmission scheme described in the following sections. DCCP provides the necessary semantics like sequence numbers and features like “Feedback/ACK vectors” are useful for loss detection. The interaction of DCCP and the application is shown in Fig. 1. DCCP congestion control is “modular”, in the sense that the protocol may easily support different congestion control mechanisms without the need of modifying its fundamental workings; for instance, both TFRC and a TCP-like mechanism have been adopted by DCCP.

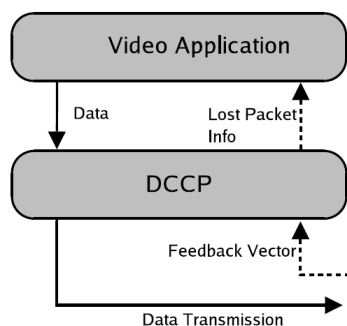


Fig.1. DCCP interaction with an application

Our WLED-ARC implementation over DCCP is an improvement over the one described in [13]. There is no need to use separate sequence numbers to detect losses for different DiffServ “colors”, used to estimate ω , because we manage all the history of packets using DCCP “ACK vectors”. The history¹ of sent packets is used to know for example if a lost packet, with a given sequence number, is *green*, *yellow* or *red*. The loss (p_c , ω , π) estimation, using exponentially weighted moving average (EWMA), is similar to the estimation method explained in [13] and is updated each time when a DCCP “ACK vector” is received.

Again, some mechanisms from TFRC, like “slow start” emulation, are also integrated and, as in Section 2.1, the slow start history is maintained separately. It should be observed that, in the Diffserv networks, the *red* packets are lost first during congestion thus, the slow start is stopped as soon as $loss_{red} > red_{threshold}$, (taken to be 40%) e.g. when lot of *red* packets are getting lost as high as 40% that means link bandwidth

is reached. Assumption is that 40% loss rate is high enough for real wireless links because connections should be simply dropped for higher loss rates.

Note that, after the slow start is stopped, for ARC or WLED-ARC, the slow start history is discarded as the loss experienced during slow start is generally high and the TCP models used do not model the slow start behaviour. The value of p_c is initialized to a value that corresponds, depending on the TCP model, to the received rate obtained by the receiver at that time.

3. The Retransmission Scheme

The entire pre-recorded video can be transmitted, in lesser time than its duration, when the network conditions are good enough. On the contrary, packets can be lost during bad channel conditions and the sending rate may not be high enough to send all the video data before its playout deadline.

The proposed selective retransmission scheme either disables or enables the retransmission of lost packets according to the current state of the network. In general the retransmission of any packet should be disabled when the network is congested. The congestion control algorithm calculates the actual sending rate to avoid congestion, while the video stream bitrate is absolutely independent of the calculated sending bitrate. When the network is in congested state or near to this state the calculated sending rate should be much lower than the video bitrate. In this situation the retransmissions should be disabled. Whereas, the retransmission should be enabled when the sending rate, determined by the congestion control protocol, is higher than the video stream rate. In this case we have to decide whether to retransmit only I-frames, or both I- and P-frames, or all the lost packets i.e., we can differentiate the lost packets according to the packet content. When the proposed sending rate is not high enough to retransmit all the lost packets, only the I-frame or I + P-frames data packets should be retransmitted.

In order to make the decision to retransmit or not, the extra load due to the retransmissions should be estimated. To determine the sending rate thresholds for the different frame type retransmissions, some statistical results on the video were used [15][16]. Usually the statistical values of the video are available in case of pre-recorded video transmission. In the case when the different frame type sizes can not be calculated before the transmission, generally acceptable values should be used to estimate the frame type size ratios in the Group of Pictures/Frames

¹ A circular buffer contains the history of packets (lost/received) with size $b_{hist} = 100$: an arbitrary value based on the empirical results. A lower b_{hist} produces oscillation in the loss estimation and a higher value makes it slow to respond to congestion.

(GOP). The statistical results show [15]² that the size of the different frame types³ can be modeled with standard normal distribution using the parameters shown in Table I.

Table I
Frame distribution

Frame type	μ (mean value)	σ (deviation)
I	5.1968	0.2016
P	3.7380	0.5961
B	2.8687	0.2675

To estimate the upper bound of the extra load due to retransmissions we analyze MPEG frame structures. The frame structure is specified by the N and M parameters. N and M deal with the intraframe to interframe coding ratio and a sequence of I, P and B frames is defined. N specifies the I frame interval whereas M determines the I or P frame interval.

Let us consider the ratio of I frame size and total GOP size as ρ_I . Similarly, the ratios for P and B frames are ρ_P and ρ_B respectively. We want to fix an upper bound on the values of these ratios. We consider 3 cases with different values of N and M as shown in Table II:

Table II
Frame ratios in MPEG streams

$N=4, M=2$	I	P	B
Number of frames	1	1	2
avg. total size in the GOP	5.1968	3.7380	5.7374
Ratio	35.42%	25.48%	39.1%
$N=16, M=2$	I	P	B
number of frames	1	7	8
avg. total size in the GOP	5.1968	26.166	22.9496
ratio	9.57%	48.17%	42.26%
$N=16, M=4$	I	P	B
number of frames	1	3	12
avg. total size in the GOP	5.1968	11.214	34.4244
ratio	10.23%	22.06%	67.71%

Table II shows the upper bounds such that the I-frame data in the GOP is estimated as 35.42% ($\rho_I=0.3542$), the P-frame data 48.17% ($\rho_P=0.4817$) and the B-frame data 67.71% ($\rho_B=0.6771$). We see that the upper bounds are $\approx 50\%$. Thus, when the statistical data is not available, we assume that I, P, B frame data in the stream to be 50%-50%-50%. Note that we are taking the upper bounds and, thus, the sum may not be

² The statistical results are more general and are not specific to H.264 coded video used in this study, but they work well for our simulations as shown in the following sections presenting the results.

³ Note that the frame size is given in number of packets and not in bytes because we will use the ratio of frame sizes.

equal to 100%. The extra load (λ_{extra}) due to the I frame retransmission is calculated as follows:

$$\lambda_{extra}^{50\%} = \rho_{50\%} \cdot r_{stream} \cdot \pi, \quad (3)$$

where r_{stream} is video stream bitrate, and π is the packet loss probability. This extra load will be used for all frame types. When the frame type size ratios are available the extra loads can be calculate for I, P or all the frame types as follows:

$$\lambda_{extra}^I = \rho_I \cdot r_{stream} \cdot \pi \quad (4)$$

$$\lambda_{extra}^{I+P} = (\rho_I + \rho_P) \cdot r_{stream} \cdot \pi \quad (5)$$

$$\lambda_{extra}^{all} = r_{stream} \cdot \pi \quad (6)$$

The frame enabling process enables or disables the retransmission, of different type of frames, according to the extra load and the sending rate given by the congestion control (CC-rate). The retransmission should be disabled when $S < r_v$, where S stands for the sending rate determined by the congestion control protocol and r_v is the video bitrate at which the video was encoded. The priority of I-frames is higher than the other frames, therefore first I-frame data packets should be retransmitted. The stipulation of retransmitting different frames is shown in Table III:

Table III
Frame type retransmissions

	General ratio ($\rho_{50\%=0.5}$)	Ratios of the actual video (ρ_I, ρ_P, ρ_B)
I frames	$r_v < S < r_v + \lambda_{extra}^{50\%}$	$r_v < S < r_v + \lambda_{extra}^I$
I+P frames	$r_v + \lambda_{extra} < S < r_v + 2\lambda_{extra}^{50\%}$	$r_v < S < r_v + \lambda_{extra}^{I+P}$
all frames	$S > r_v + 2\lambda_{extra}^{50\%}$	$r_v < S < r_v + \lambda_{extra}^{all}$

4. Simulation Results

In order to look at the performance of the different congestion control based selective retransmission scheme and the proposed retransmission thresholds, we analyze some scenarios with ns-2 [17] network simulator. We use dumbbell topology with $1Mbps$ links and $10ms$ link delays. The “wireless link” is the bottleneck link, in order to introduce wireless packet losses using a simple random drop model with the given loss probability. We study two types of cases; one with the Droptail queue management and other with RIO AQM in the bottleneck link.

4.1. Droptail

First we examine how the different congestion control based selective retransmission schemes can utilize the available bandwidth on the $1Mbps$ bottleneck link. Either FTP or WWW background traffic are set up during the simulations. The network delay is $40ms$, while the Droptail queue limit is set to 40 . To analyze the quality of the $360kbps$ H.264 video stream (reference video is “mother and daughter”), the PSNR (Peak Signal to Noise Ratio) objective quality parameter is used.

The decision algorithm for the retransmissions scheme is determined by the actual calculated sending rate given by the congestion control algorithm. The TFRC sending rate is the lowest; therefore using this protocol for the selective retransmission disables the retransmission when ARC and WLED-ARC still enables it, especially in the case of high wireless loss probability.

The retransmissions and the used congestion control method have significant impact on the number of packets lost. Slightly less packets are lost when the retransmission is disabled although the video quality is the worst as will be shown later in this section. Using ARC as congestion control protocol the high sending rate can cause higher loss probability, but more packets can be retransmitted improving the video quality.

When the difference between the video bit-rate and the determined sending rate is small, only the important data packets should be retransmitted. Retransmitting all the lost packets causes congestion in the network. If more packets are retransmitted than the available link capacity, the packets will be lost again. The ARC protocol efficiently approaches the link capacity limits and, therefore, the differentiation between the packets is beneficial. We investigate the number of lost packets when packet differentiation and retransmission thresholds are used. As Fig. 2 illustrates, the number of lost packets is 3-5% less when differentiation is used. The impact of different frame types on the video quality is not the same; therefore our motivation is to protect the important frames like I pictures to avoid the propagation of errors. When there is not enough link capacity to retransmit all the packets, only the I frame data packets or I and P frames are retransmitted.

According to the next simulation results, where we compare different congestion control schemes, the best video quality is achieved with ARC and ARC with frame differentiation. Moreover, a slight improvement can be seen when using frame differentiation in addition to ARC. With these methods more lost

packets can be successfully retransmitted. As shown⁴ in Fig. 3, the Peak Signal to Noise Ratio (PSNR), representing video quality, is significantly higher when ARC is used and the wireless loss probability is high.

This is because ARC does not back-off unnecessarily during wireless losses whereas, TFRC does that and the sending rate available to the video application is very low.

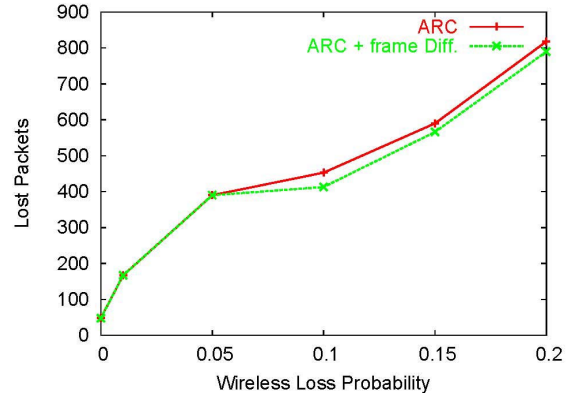


Fig. 2. Lost packets with differentiation of frames and no background traffic.

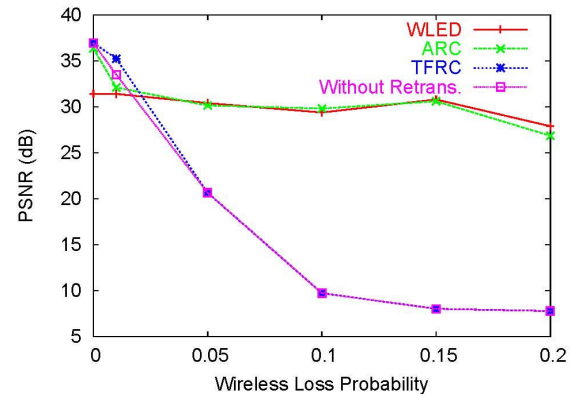


Fig. 3. Avg. PSNR with 1 FTP in background.

With TFRC, the video application not only fails to retransmit the lost data, but also it is not able to send the complete video sequence because of the low value of the sending rate.

We have also compared the video quality when the retransmission threshold is determined by general frame type size ratios ($\rho_{50\%}$) and by actual ratios (the analyzed H.264 video frame size ratios are: $\rho_I=18\%$, $\rho_P=60\%$, $\rho_B=22\%$). Fig. 4 shows that the difference is only $0.1-0.4dB$ in the case when the valid statistical

⁴Note that the “without retransmission” case corresponds to the case when video application uses TFRC scheme as the congestion control scheme and it never retransmits the lost video data.

values (possible for pre-recorded video streaming) are used for the retransmission threshold determination. Thus, our default values can be used without significant degradation in the video quality, when the valid statistical values are not available.

In order to analyze the effect of the background load, WWW connections are used on the shared bottleneck link. The number of lost packets increases on increasing the load on the bottleneck link, although the wireless loss probability is constant ($\omega = 10\%$, i.e., high). Fig. 5 shows that ARC performs significantly better than the other schemes. This is because when ω is high the other schemes can not differentiate between congestion and wireless losses and the sending rate is very low. Not only they disable the retransmission, but they also send the video data at a rate significantly lower than the video bit-rate.

4.2. RIO (Red In and Out)

The retransmission method is investigated in a Diffserv network as well, where the important packets are marked as green and the least important packets as red. Moreover, it is possible to use WLED-ARC as it assumes a Diffserv network. In the simulations, the frame differentiation is also used to prefer the retransmission of the most important I frames. Again, as shown in Fig. 6, the WLED and ARC schemes perform significantly better than TFRC and TFRC without retransmissions.

The increase in the background traffic has no effect on the TFRC-based transmission when the wireless loss is already high because the determined sending rate is already too low to enable retransmission. In the case of TFRC, the retransmission is disabled because, due to the high loss rate (10%), the calculated sending rate is too low to even transmit the video at its video bit-rate. This is the reason that the TFRC retransmission and the case without retransmission are very similar in the next case when we fix the wireless loss to $\omega = 10\%$. The results are shown in Figures 7 and 8. The performance of ARC- and WLED-ARC-based retransmission is significantly better in wireless networks with high loss ratio. The quality improvement is due to the large number of retransmitted packets that are lost when the retransmission is enabled. ARC and WLED-ARC effectively forward the video stream, and retransmit the lost data as shown in Fig. 8, achieving significant improvement in the video quality.

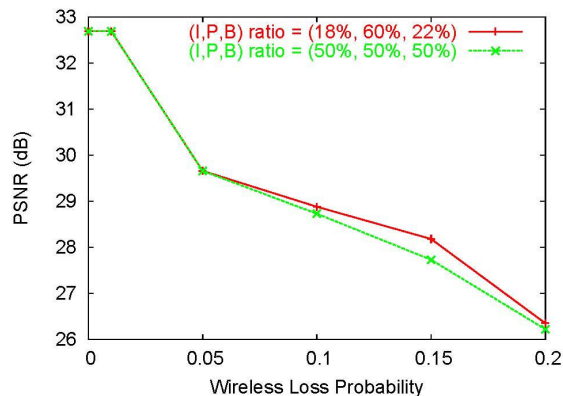


Fig. 4. Avg. PSNR with different threshold methods and 1 FTP in background.

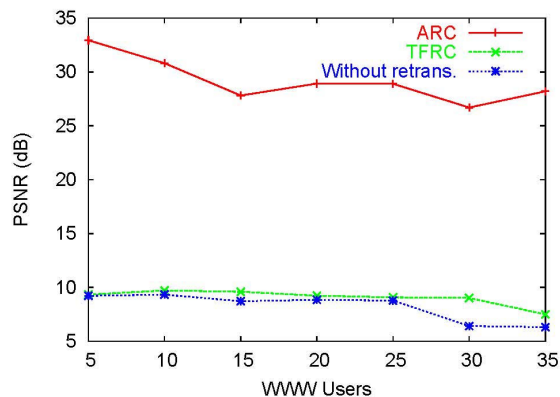


Fig. 5. Avg. PSNR with WWW users as background traffic and $\omega = 0.1$.

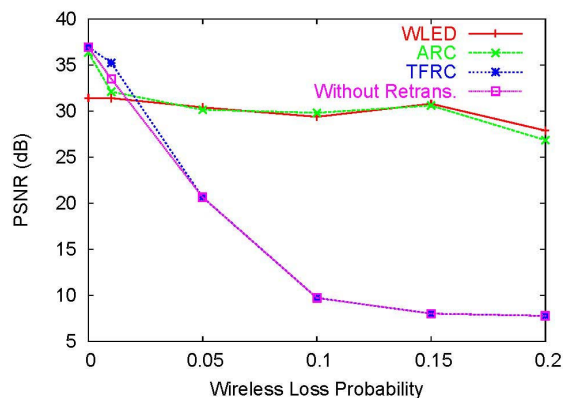


Fig. 6. Avg. PSNR in a Diffserv network with varying ω and 10 WWW users.

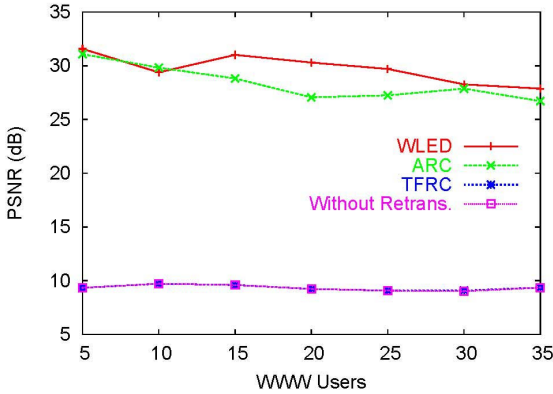


Fig. 7. Avg. PSNR in a Diffserv network with $\omega = 0.1$.

5. Conclusion

Packet losses, due to bad radio conditions in wireless networks, not only degrade the video quality but, render the current congestion control algorithms that back-off on every loss, as inefficient.

In order to counter this problem, we integrate the congestion control schemes with an adaptive retransmission scheme in order to selectively re-transmit some lost video packets. We, also, integrate a wireless loss estimation scheme to avoid the confusion between two types of losses and improve the efficiency. Our results show that the integrated scheme not only improves the video quality due to the re-transmission and recovery of some of the lost video data but, the recovery itself is efficient because of the wireless loss estimation.

Acknowledgement

This work was supported by the ANEMONE project which is partly funded by the Sixth Framework Program of the European Commission's Information Society Technology.

References

[1] M. Feamster, H. Balakrishnan, "Packet Loss Recovery for Streaming Video", 12th International Packet Video Workshop, Pittsburgh, PA, Apr. 2002.
 [2] A. Miyazaki, A. H. Fukushima, K. Hata, T. Wiebke, R. Hakenberg, C. Burmeister, Matsushita, "RTP payload formats to enable multiple selective retransmission", IETF, draft-ietf-rtsp-selret-04.txt, Nov. 2001.
 [3] B. Zheng, M. Atiquzzaman, "Network Requirement for Management of Multimedia over Wireless Channel", LNCS Vol. 2496, 5th IFIP/IEEE International Conference on Management of Multimedia Networks and Services, London, UK, 2002.

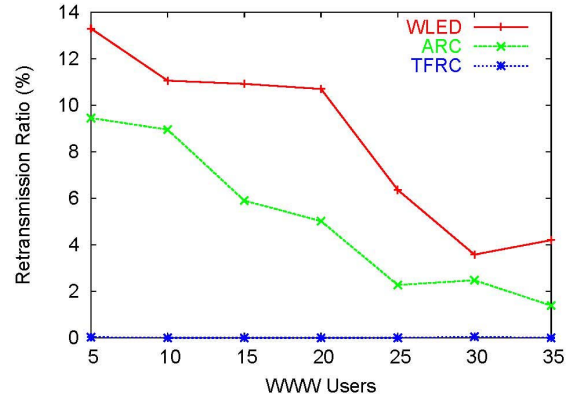


Fig. 8. Ratio of no. of retransmitted packets and no. of lost packets with $\omega = 0.1$.

[4] M. Piecuch, K. French, G. Oprica, M. Claypool, "A Selective Retransmission Protocol for Multimedia on the Internet", Proceedings of SPIE International Symposium on Multimedia Systems and Applications, Boston, MA, USA Nov. 2000.
 [5] H. Hagino, Y. Miyazaki, Y. Onoe, Y. Atsumi, H. Komaki, M. Taniguchi, N. Yamanouchi, "A Playout Time Oriented Retransmission Scheme for Multimedia Streaming Systems", HSNMC 2003, Estoril, Portugal, 23-25 July 2003.
 [6] K.-J. Grinnemo, J. Garcia, and A. Brunstrom, "Taxonomy and survey of retransmission-based partially reliable transport protocols", Computer Communications, vol. 27, no. 15, pp. 1441-1452, Sept. 2004.
 [7] Kohler, Handley, Floyd, "Datagram Congestion Control Protocol", Internet Engineering Task Force, Mar. 2006, RFC 4340.
 [8] E. Kohler, M. Handley, S. Floyd, "Designing DCCP: Congestion Control Without Reliability", SIGCOMM 2006, Pisa, Italy, 11-15 Sept. 2006.
 [9] S. Floyd, M. Handley, J. Padhye, and J. Widmer "Equation based congestion control for unicast applications", In Proceedings of ACM SIGCOMM 2000, pages 43-56, Stockholm, Aug. 2000.
 [10] S. Floyd, E. Kohler, and J. Padhye, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)", RFC 4342, IETF, Mar. 2006.
 [11] M. Handley, S. Floyd, J. Padhye, and J. Widmer. TCP Friendly Rate Control (TFRC): Protocol Specification. Internet Standards Track RFC 3448, IETF, Jan. 2003.
 [12] O. B. Akan and I. F. Akyildiz, "ARC: the analytical rate control scheme for real-time traffic in wireless networks," IEEE/ACM Transactions on Networking, vol. 12, no. 4, pp. 634-644, 2004.
 [13] K. D. Singh, D. Ros, L. Toutain and C. Viho. "Improving Multimedia Streaming over wireless using end-2-end estimation of Wireless Losses", IEEE 64th Vehicular Technology Conference, Montreal, Canada, Sept. 2006.
 [14] R. Makkar, I. Lambadaris, J. Salim, N. Seddigh, B. Nandy, and J. Babiarz, "Empirical study of buffer management scheme for DiffServ Assured Forwarding PHB", IEEE ICCCN 2000.
 [15] M. Krantz and H. Hughes, "A Traffic Model for MPEG-Coded VBR Streams", Proceedings of the ACM SIGMETRICS Conference on the Measurement and Modeling of Computer Systems, 1995.
 [16] D. Isovich, G. Fohler, L. Steffens: "Real-time issues of MPEG-2 playout in resource constrained systems", IEEE Journal of Embedded Computing, Issue 3, June 2004
 [17] Ns-2 - Network Simulator, <http://www.isi.edu/nsnam/ns/index.html>