

Feedback-Free Packet Loss Recovery for Video Multicast

Xing Zheng S.-H. Gary Chan

Department of Computer Science

The Hong Kong University of Science and Technology

Clear Water Bay, Kowloon, Hong Kong

Email: {cszx, gchan}@cs.ust.hk

Qian Zhang Wen-Wu Zhu Ya-Qin Zhang

Microsoft Research Asia

5F, Beijing Sigma Center, No. 49, ZhiChun Road

Haidian District, Beijing 100080, P. R. China

Abstract—In video streaming over multicast networks, error recovery is essential to alleviate the effect of packet loss. In this paper, we study a feedback-free recovery scheme which combines the strength of FEC (namely, a parity packet can repair any lost packet) and pseudo-ARQ (namely, incremental recovery). To account for the receiver heterogeneity, the server multicasts layered video streams for the receivers to join. For each layer, the receivers may join dynamically additional multicast channels of FEC and pseudo-ARQ packets to recover local losses. In order to offer quality video, we address the following two issues: 1)“Menu creation”: Given a certain maximum error rate after correction (i.e., residual error rate), what is the combination of FEC and pseudo-ARQ packets for the server to send so as to minimize a target receiver’s bandwidth; and 2)“Menu selection”: Given the server’s menu, what’s the combination of FEC and pseudo-ARQ packets for the receiver to join so as to minimize its residual error rate given its local loss probability, bandwidth and loss pattern. We present the analysis of the scheme and show that our scheme can substantially reduce a receiver’s residual error rate as compared with pure FEC or pure pseudo-ARQ alone (by cutting it as much as half).

ARQ [5], [6] — In ARQ, the receivers feedback the sequence numbers of the lost packets back to the server, so that the server retransmits those packets to the receivers. Since the retransmitted packets are original packets, ARQ approach achieves “incremental recovery,” where each retransmitted packet, if successfully received, can be used for incremental quality improvement. This is opposite to FEC, where either all l lost data packets are recovered or the FEC packets are received in vain. Therefore, such approach performs well when the loss probability is high. However, the ARQ-based approach has two disadvantages: long recovery delay and feedback implosion [7], [8]. Although much work has been done in the literature to address these two issues, most of it is relevant and applicable for reliable multicast rather than video streaming where there is a stringent recovery deadline [9], [10], [11].

We propose a simple feedback-free scheme combing the strengths of FEC and ARQ. Some of the recovery packets sent out from the server are original source packets. These original source packets look like the retransmitted packets with a certain delay in the receiver point of view. They are hence called “pseudo-ARQ” packets in this paper. Note that some previous work uses pseudo-ARQ packets in their schemes to refer to “delayed” FEC packets instead of the *original source packets* as in our scheme. The reason we use original source packets is due to their characteristics of “incremental recovery”. For example, a receiver subscribes 5 recovery packets to recover 3 lost packets and it only get 2 out of 5 packets. If recovery packets are all FEC packets, the receiver’s effort is in vain. However, if they are original source packets, the receiver at least gets 2 packets.

Our video multicast system with error recovery consists of two important components:

1. Server: In order to provide scalable video quality for heterogeneous receivers, the server encodes the video sequence into multiple layers using some scalable encoding techniques [12]. These layers are sent to different multicast groups over the networks. To recover packet loss, error recovery packets (such as FEC or pseudo-ARQ packets) are introduced to each layer of source packets and are multicast in different groups. In this work, we focus on an arbitrary layer of video and its associated recovery packets. The server generates $N - K$ error recovery packets for every K source packets. The $N - K$ recovery packets consist of some FEC parity packets and some pseudo-ARQ packets, i.e., “menu” for the client to select to recover their packet loss. All recovery packets are multicast into a number of multicast groups after the server sends the K source packets. An important issue of the server is hence how to “create the menu” in terms of its composition and sending mechanism so as to minimize the receivers’ overall error rates after correction, the so-called “residual” error rate. In order to offer satisfactory video quality, the residual error rate must not exceed a certain value (say 2 – 5%).

2. Clients: The clients join the corresponding multicast groups to receive desirable data and recovery packets according to their bandwidth and local loss probabilities. Receivers in the multicast group are heterogeneous. The packet loss probability of a receiver

I. INTRODUCTION

Recently, there has been growing interest in video streaming over networks. Such services include movie broadcasting, distance learning, video conferencing, etc. These services in general involve point to multi-point communication, i.e., a video sequence stored or generated at a server is delivered to a group of receivers over the networks [1]. (In this paper, we use “client” and “receiver” interchangeably.) In video transmission over networks, packet loss is inevitable. In order to provide video of good quality, end-to-end loss rate has to be kept below a certain value. Hence, error recovery schemes are essential to guarantee good services [2].

Traditional approaches for error recovery can be divided into two categories — FEC (Forward Error Correction) and ARQ (Automatic Repeat Request):

FEC [3], [4] — FEC combines redundant information with video sources using, for example, the Reed-Solomon correction code. In FEC, the server interleaves $T - K$ FEC packets with K source packets. The receiver can recover the K source packets as long as it receives any K out of the T packets. Since FEC packets do not associate with any source packet, it can help to recover any lost packet at the receivers. Moreover, FEC approach is feedback-free and hence avoids the issues of long recovery delay and feedback implosion. However, FEC has some weakness. It has to be designed for the worst case and hence the system is usually “overprotected” in the normal conditions. Furthermore, if the receiver loses l packets out of K data packets, it must receive at least l FEC packets to recover l lost data packets. Otherwise, the receiver can recover none of l lost packets. This is the so-called “all-or-none error recovery.” From the above, we see that FEC-based approach is not effective when the loss probability is high.

This work is supported, in part, by the Competitive Earmarked Research Grant from the Hong Kong Research Grant Council (HKUST6014/01E and HKUST6199/02E) and Sino Software Research Institute at the HKUST (SSRI01/02.EG21).

may vary from nearly zero to tens of percents and may vary over time. In addition, receivers may have different end-to-end bandwidth, depending on the number of source and recovery streams they join. Since both the loss probability and bandwidth vary over time, clients may join different recovery streams to recover local loss. It is well-known that there is a tight time constraint for video playing, within which the receiver needs to recover as many lost packets as possible. This constraint implies a maximum number of source data and recovery packets the receiver can obtain given a certain end-to-end bandwidth. An important issue is how to minimize the receiver's residual error rate given its loss probability and bandwidth constraint by choosing the appropriate components of recovery packets from the server's menu. This is the so-called "menu selection" problem.

The strengths of our scheme are: 1) Fast recovery: Because the pseudo-ARQ and FEC packets are sent from the server directly right after the K source packets, the receiver can recover the lost packets without the long recovery delay so that it can play the video on time; 2) Feedback-free: Since the pseudo-ARQ packets are not sent given the feedbacks from the receivers, our hybrid scheme is feedback-free and do not suffer from the request implosion issue; 3) Autonomous selection: The receiver can autonomously subscribe the desired recovery packets to minimize their residual error rate without their interactions with each other.

Given the scheme, we address the following two important issues: 1) Server's menu creation: At the server end, what is the optimal combination of FEC and pseudo-ARQ packets to minimize the target receiver's bandwidth so that its residual error rate is kept below a certain value? 2) Receiver's menu selection: At the receiver, what is the optimal combination of FEC packets and pseudo-ARQ packets to minimize its residual error rate given its local loss pattern, bandwidth and loss probability?

In this paper, we formulate the optimization problems and analyze the error recovery system. Our numerical results show that the hybrid scheme can substantially reduce the receiver's residual error rate as compared with FEC or pseudo-ARQ alone.

This paper is organized as follows. We describe the system and present its analysis in Section II. In Section III, we show some illustrative numerical results and conclude in Section IV.

II. SYSTEM DESCRIPTION AND ANALYSIS

A. System Description and Problem Formulation

In our system, the server sends out packets in blocks and the transmission time is divided into slots of fixed length. Each block consists of K source packets and associated $N - K$ recovery packets. Without loss of generality, we consider the transmission of packets belonging to block I for a layer. The other layers are transmitted in parallel similarly.

We show the server's menu at time slot I and $I + 1$ in Fig. 1. At time slot I , the server sends K source packets labelled $(I, 1)$, $(I, 2)$, ..., (I, K) . The $N - K$ error recovery packets corresponding to block I are sent in time slot $I + 1$. These $N - K$ error recovery packets consist of N_A pseudo-ARQ packets and N_F FEC packets, where $N_A + N_F = N - K$. The pseudo-ARQ packets consists of M copies for each source packet, where $M \geq 1$ is an integer. Clearly, $N_A = M \cdot K$.

Note that a receiver's bandwidth limits the maximum number of packets that it can receive within the delay constraint. We define the receiver's bandwidth n as the total number of packets that it can receive in one time slot. Since the data packets in the current time slot cannot be replaced by recovery packets to recover errors in the previous slots, we consider a "constant bit-rate client", or a CBR-client for short, which gets a fixed number of recovery packets $n - K$ in each time slot.

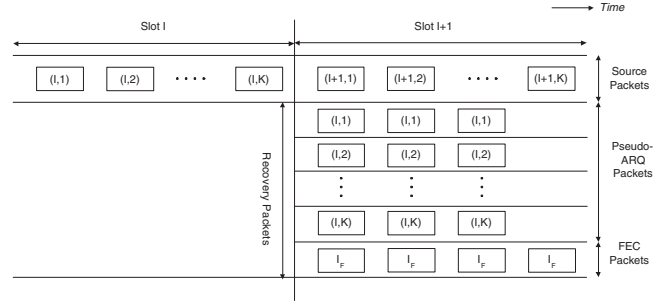


Fig. 1. Server's Menu.

Given the server's menu, we consider a receiver with loss probability p and bandwidth n . (In practice, a target receiver is associated with loss probability \hat{p} , which may be the maximum loss probability the receivers are likely to experience.) At time slot I , the receiver subscribes the K source packets. Out of these K source packets, the number of lost packets l ($l \leq K$) is random and may vary from slot to slot due to non-stationary loss probability. We consider the recovery decision is made at the slot boundary, i.e., at the end of the slot I , the receiver chooses $n_F(l)$ FEC packets and $n_A(l)$ Pseudo-ARQ packets (both of them functions of l) so that the overall residual error rate, denoted as ε , is no more than ε_0 , i.e., $\varepsilon = \sum_{l=0}^K \binom{K}{l} (1-p)^{K-l} p^l \cdot \varepsilon(l) \leq \varepsilon_0$. Some of $n_F(l)$ FEC packets or $n_A(l)$ pseudo-ARQ packets may also be lost during their transmission in time slot $I + 1$. At the end of the slot $I + 1$, the receiver does error recovery based on the source packets received in the slot I and the recovery packets received in the slot $I + 1$. Hence the delay of this scheme is 2 time slots. Here we have ignored the join and leave latency. If join or leave latency is taken into considerations, n would be correspondingly reduced.

Note that the receiver may subscribe multiple copies of its lost packets, i.e., $n_A \geq l$. To minimize the residual error rate, the number of copies for each lost packet clearly should be more or less the same, i.e., let $m_1 = \lfloor n_A/l \rfloor$ and $m_2 = \lceil n_A/l \rceil$, then the receiver gets m_2 copies of the first $n_A - m_1 l$ lost packets and m_1 copies of the rest lost packets.

Given the system operation, we hence have the following menu "creation" and "selection" problems:

1. Server's menu creation: At the server end, the server creates a menu for a target receiver with a loss probability \hat{p} . The objective is to minimize a target receiver's bandwidth such that the target receiver's residual error rate $\varepsilon \leq \varepsilon_0$. Given a certain number of lost packets l for the target receiver, we can always find a family of $(n_F(l), n_A(l))$ pairs to meet the loss constraint, where $n(l) = K + n_F(l) + n_A(l)$. We hence seek the minimum server bandwidth given by $N = K + N_F + N_A$, where the number of FEC packets at the server N_F is given by

$$N_F = \max_l n_F(l). \quad (1)$$

and the number of copies of pseudo-ARQ packets N_A is given by

$$N_A = K \cdot M = K \cdot \max_l \left\lceil \frac{n_A(l)}{l} \right\rceil. \quad (2)$$

Thus the optimal problem of server's menu creation may be formulated as:

$$\begin{aligned} \text{Given:} & \quad \hat{p} \\ \min_{\{l: n_F(l), n_A(l)\}} & \quad n \\ \text{such that:} & \quad \varepsilon \leq \varepsilon_0 \end{aligned}$$

TABLE I
NOMENCLATURE USED IN THIS PAPER.

K	: the number of source packets in one block
\hat{p}	: the target packet loss probability for the server
p	: packet loss probability of a receiver
l	: the number of lost packets out of K source packets
G	: the number of useful data after correction at the receiver
ε	: a receiver's residual error rate ($\varepsilon = \frac{K-E[G]}{K}$)
ε_0	: the error rate constraint
N_A	: the number of pseudo-ARQ packets in one block from the server
N_F	: the number of FEC packets in one block from the server
N	: the total number of packets in one block from the server ($N = K + N_A + N_F$)
n_A	: the number of pseudo-ARQ packets subscribed in one block at the receiver
n_F	: the number of FEC packets subscribed in one block at the receiver
n	: the total number of packets subscribed in one block at the receiver ($n = K + n_A + n_F$)

2. Receiver's menu selection: Given the server's menu (N_F, M) , a receiver with loss probability p , bandwidth n and the number of lost packet l has to determine $n_F(l)$ and $n_A(l)$. Because different pairs of $(n_F(l), n_A(l))$ may result in different ε , the receiver should choose an optimal pair $(n_F(l)^*, n_A(l)^*)$ so as to minimize its ε . Clearly, the receiver can only select as many packets as the server's menu provides, i.e., $n_F(l) \leq \min(N_F, n - K)$ and $n_A(l) \leq \min(Ml, n - K)$. Then the receiver can search the pair $(n_F(l), n_A(l))$ which leads to the minimal ε and choose that as its optimal selection. The optimal problem of receiver's menu selection can hence be posed as:

$$\begin{aligned} \text{Given:} & \quad N_F, M, p, n, l \\ \min_{\{n_F, n_A\}} & \quad \varepsilon \\ \text{such that:} & \quad n_F(l) \leq \min(N_F, n - K); \\ & \quad n_A(l) \leq \min(Ml, n - K). \end{aligned}$$

B. System Analysis

We address via analysis the problems of server's menu creation and receiver's menu selection. The main issue is to calculate ε . For server's menu creation, we need to calculate the ε for a target receiver with loss probability \hat{p} . The calculation of ε is the same for that of an arbitrary receiver with loss probability p in receiver's menu selection. Hence we only show how to calculate ε for an arbitrary receiver. (A target receiver's residual error rate ε can be calculated by replacing p by \hat{p} .) Some of the important nomenclature used in this paper are shown in Table I.

We consider each packet is lost with probability p , independent of each other. Therefore, the number of received packets in a slot is a binomial random variable. Let Q and F be the random variables of the pseudo-ARQ and FEC packets received ($Q \leq n_A(l)$, $F \leq n_F(l)$) and G_l be the number of useful packets after correction.

Recall that the error rate ε given l is denoted as $\varepsilon(l)$. Clearly,

$$\varepsilon(l) = \frac{K - E[G_l]}{K}. \quad (3)$$

For a CBR-client, the residual error rate is given by

$$\varepsilon = \sum_{l=0}^K \binom{K}{l} (1-p)^{K-l} p^l \cdot \varepsilon(l) \quad (4)$$

and we seek $\varepsilon \leq \varepsilon_0$. Therefore, what remains to be found is $E[G_l]$.

Clearly, given that l source packets are lost in slot I , G_l is given by

$$G_l = \begin{cases} K, & \text{if } Q + F \geq l; \\ K - l + Q, & \text{otherwise.} \end{cases} \quad (5)$$

The expected value of G_l is then given by

$$\begin{aligned} E[G_l] &= K \cdot P(Q + F \geq l) + (K - l + E[Q|Q + F < l]) \\ &\quad \cdot P(Q + F < l) \\ &= K + (E[Q|Q + F < l] - l) \cdot P(Q + F < l). \end{aligned} \quad (6)$$

What is left is to find $E[Q|Q + F < l]$ and $P(Q + F < l)$. Here we consider two cases: $n_A < l$ and $n_A \geq l$.

If $n_A < l$, the receiver subscribes at most a single copy of pseudo-ARQ packets for each lost packet. The expected number of received pseudo-ARQ packets is given by

$$\begin{aligned} E[Q|Q + F < l] &= \sum_{q=0}^{n_A} q \cdot P(Q = q|Q + F < l) \\ &= \frac{\sum_{q=0}^{n_A} q \cdot P(Q = q)P(F < l - q)}{P(Q + F < l)}. \end{aligned} \quad (7)$$

The probability that the number of received recovery packets is less than the lost packets is given by

$$\begin{aligned} P(Q + F < l) &= \sum_{q=0}^{n_A} P(Q = q) \cdot P(F < l - q) \\ &= \sum_{q=0}^{n_A} \binom{n_A}{q} (1-p)^q p^{n_A - q} \\ &\quad \times \left[\sum_{f=0}^{l-q-1} \binom{n_F}{f} (1-p)^f p^{n_F - f} \right]. \end{aligned} \quad (8)$$

If $n_A \geq l$, the receiver may subscribe multiple copies of pseudo-ARQ packets for each lost packet. We let the number of copies for each lost packet be m , where $m = n_A/l$. (Here we assume that m is a real number. We also do exact analysis using two integers m_1 and m_2 . The experiment results do not differ much. Therefore, we conclude that it is reasonable to use a real m in the analysis for simplicity.) To recover a lost packet using pseudo-ARQ packets, at least one out of the m copies has to be received. Clearly,

$$\begin{aligned} E[Q|Q + F < l] &= \sum_{q=0}^{l-1} q \cdot P(Q = q|Q + F < l) \\ &= \frac{\sum_{q=0}^{l-1} q \cdot P(Q = q)P(F < l - q)}{P(Q + F < l)} \end{aligned} \quad (9)$$

and

$$\begin{aligned} P(Q + F < l) &= \sum_{q=0}^{l-1} P(Q = q) \cdot P(F < l - q) \\ &= \sum_{q=0}^{l-1} \binom{l}{q} (1-p^m)^q p^{m(l-q)} \\ &\quad \times \left[\sum_{f=0}^{l-q-1} \binom{n_F}{f} (1-p)^f p^{n_F - f} \right]. \end{aligned} \quad (10)$$

The above formulas are also applicable for pure schemes. The expected value of G_l for pure-FEC scheme or for pure pseudo-ARQ

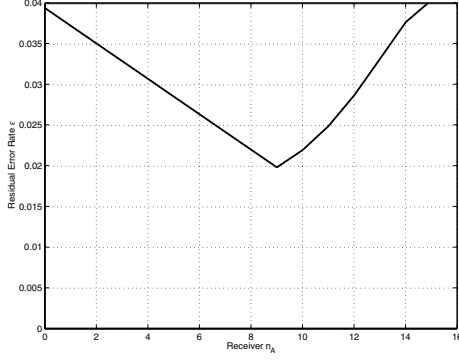


Fig. 2. A client's optimal selection ($K = 30$, $\hat{p} = 0.4$, $n = 45$, $l = 9$).

scheme can be obtained by substituting $n_A = 0$ or $n_F = 0$. In pure FEC scheme, $E[Q|Q + F < l] = 0$. Obviously,

$$E[G_l] = K - l \cdot \sum_{f=0}^{l-1} \binom{n_F}{f} (1-p)^f p^{n_F-f}. \quad (11)$$

In pure pseudo-ARQ scheme, $G_l = K - l + Q$. Therefore,

$$E[G_l] = \begin{cases} K - l + n_A(1-p), & \text{if } n_A < l; \\ K - l \cdot p^m, & \text{otherwise.} \end{cases} \quad (12)$$

III. ILLUSTRATIVE RESULTS

In this section, we show some illustrative results for server's menu creation and receiver's menu selection. As a baseline, we use $\varepsilon_0 = 4\%$ and $K = 30$. (We vary K in the range $[5, 50]$ in experiments and the results increase linearly when K increases. Therefore, we can simply choose $K = 30$.)

A. Server's menu creation

For server's menu creation, we use a baseline with $\hat{p} = 0.4$, $n = 45$ and $l = 9$. (The minimum bandwidth requirement for the target receiver with $\hat{p} = 0.3$ to satisfy the residual error rate constraint is $n = 45$.) In Fig. 2, we plot ε versus n_A for a client given a certain number of lost packets $l = 9$. Note that the case $n_A = 0$ corresponds to the pure FEC case while the case $n_A = n - K = 15$ corresponds to the pure pseudo-ARQ cases. As n_A increases, ε first decreases and then increases. It shows that the trade-off between selecting FEC packets and selecting Pseudo-ARQ packets. When l is small, subscribing pseudo-ARQ packets is better than subscribing FEC packets because the receiver can avoid all-or-none problem of FEC packets. Conversely, subscribing multiple copies of pseudo-ARQ packets does not help much more because the receiver may get multiple copies of a certain lost data packet. Clearly, there is an optimal (n_A^*, n_F^*) to achieve the minimum ε . Such optimum is a mix of pseudo-ARQ and FEC packets and can achieve a substantial reduction in ε .

Figure 3 shows the server's menu for the target receiver against p . When p increases, the two components of the server's menu N_A and N_F increase as well due to high protection to lost packets. Note that N_A is an integer multiple of K , i.e., $N_A = 30, 60, \dots$ while N_F increases more smoothly. That is due to the fact that the number of pseudo-ARQ packets sent by the server should be an integer multiple of K source packets, i.e., $N_A = K \cdot M$, where M is an integer. Therefore, there are a certain degree of redundancy in the server's menu creation.

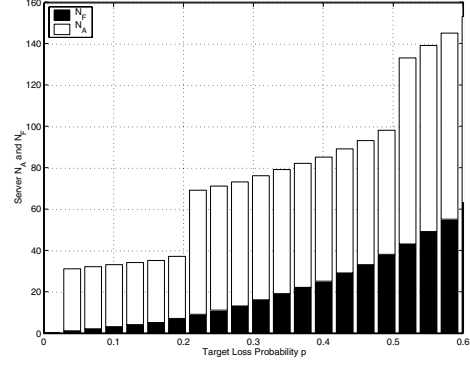


Fig. 3. The server's menu N_A and N_F ($K = 30$, $\varepsilon_0 = 4\%$).

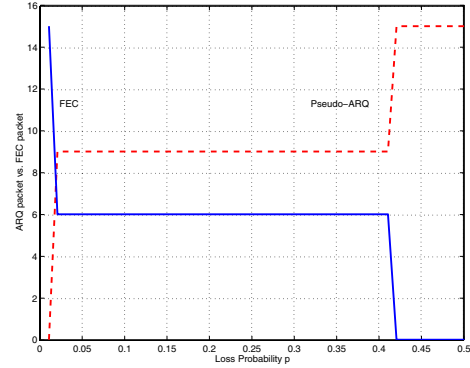


Fig. 4. A client's menu selection ($K = 30$, $\varepsilon_0 = 4\%$, $\hat{p} = 0.4$, $M = 2$, $N_F = 25$, $n = 45$, $l = 9$).

B. Receiver's menu selection

For receiver's menu selection, we use a baseline with $\hat{p} = 0.4$, $M = 2$, $N_F = 25$, $n = 45$ and $l = 9$. The server's menu ($M = 2$, $N_F = 25$) is designed for the target receiver with $\hat{p} = 0.4$. In Fig. 4, we plot the receiver's optimal selection of n_A and n_F versus p given the baseline. When p is low, the receiver only subscribes FEC packets for recovery. As p increases, the receiver subscribes both FEC and pseudo-ARQ packets. Finally, when p is very high, the receiver subscribes as many pseudo-ARQ packets as it can. (Since the server's menu only provide two copies of pseudo-ARQ packets, i.e., $M = 2$, the receiver can subscribe at most $M \times l = 2 \times 9 = 18$ pseudo-ARQ packets). This also indicates that the pure FEC scheme is more effective when p is low while the pure pseudo-ARQ scheme is more effective when p is high.

We next show in Fig. 5 a comparison of pure FEC, pure pseudo-ARQ and hybrid schemes. Given $\hat{p} = 0.4$, each scheme creates its own optimal server's menu respectively. Given the server's menu of each scheme, the client chooses recovery packets so as to minimize its ε . In this figure, we plot a client's ε of different schemes versus p given $n = 45$. As p increases, ε increases. When p is low, the pure FEC scheme is more effective than the pure pseudo-ARQ scheme while pure pseudo-ARQ scheme performs better than pure FEC scheme when p is high. This indicates the trade-off of using the pure FEC scheme or the pure pseudo-ARQ scheme. Furthermore, ε of the hybrid scheme is always lower than that of the pure FEC scheme or the pure pseudo-ARQ scheme. Especially when p is high, our hybrid scheme can substantially reduce the residual error rate. In this example, when p is about 0.25, ε is cut by almost half. This shows the strength of our hybrid scheme as compared with using FEC or pseudo-ARQ scheme alone.

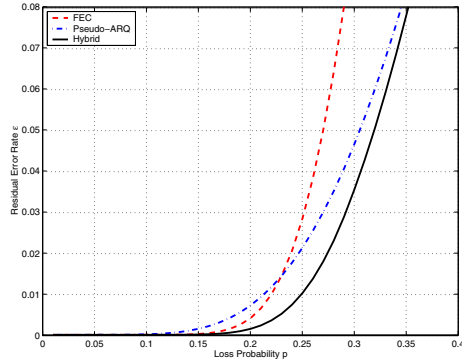


Fig. 5. A client's residual error rates for pure FEC, pure pseudo-ARQ and hybrid schemes ($K = 30$, $\varepsilon_0 = 4\%$, $\hat{p} = 0.4$, $n = 45$).

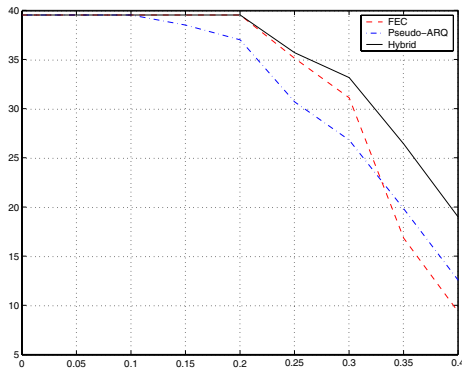


Fig. 6. A client's PSNR of the video transmitted using pure FEC, pure pseudo-ARQ and hybrid schemes ($K = 30$, $\varepsilon_0 = 4\%$, $\hat{p} = 0.4$, $n = 45$, $GoP = 8$, $QP = 3$).

We also compare the three schemes by streaming a QCIF foreman sequence encoded with H. 263+, $GoP = 8$ and $QP = 3$. The PSNR values of each frame of the decoded video sequences transmitted using different schemes versus p are showed in Fig. 6. When p is small, three schemes have the same PSNR value since there is no residual error rates. When p increases, the PSNR values of three schemes decrease, which mean the video quality becomes worse. Moreover, the FEC scheme performs better than pseudo-ARQ when p is small, while vice versa when p becomes higher. Note that the PSNR values of the hybrid scheme are always better than the other two schemes. These results are consistent with the results from the numerical experiments.

Besides, we show the subjective performances of two schemes in Fig. 7. Note that the visual quality of the video transmitted using the FEC scheme is worse than that of the hybrid scheme since the video suffers higher residual error rate.

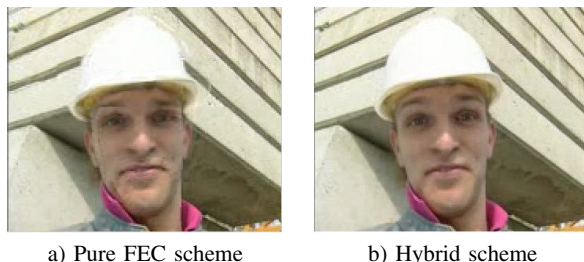


Fig. 7. Subjective performance of a video with two schemes.

IV. CONCLUSIONS

In this paper, we have studied the error recovery system for video streaming over multicast networks. The main issues in the system are how the server creates a menu to send recovery packets so as to minimize the overall receiver's residual error rate and how an arbitrary receiver optimally selects recovery streams to join in order to achieve better video quality, the so-called server's menu creation and receiver's menu selection. Note that both FEC and ARQ approaches have weakness, we propose a feedback-free error recovery scheme. In this scheme, the server sends out optimal combination of FEC and pseudo-ARQ packets without feedback so that the receivers can recover packet loss quickly and autonomously.

In our system, we describe server's menu creation and receiver's menu selection for CBR-clients. Analysis approach is adopted to address these two issues. The numerical results show that the illustrative examples of the optimal server's menu creations and optimal receiver's selection. Besides, by comparing three error recovery schemes, i.e., pure FEC, pure pseudo-ARQ and hybrid schemes, our hybrid scheme can perform better than pure FEC or pseudo-ARQ schemes. It can cut the receiver's residual error rate substantially (by half).

REFERENCES

- [1] S. Deering, "Internet multicast routing: State of the art and open research issues," Multimedia Integrated Conferencing for Europe (MICE) Seminar at the Swedish Institute of Computer Science, Stockholm, Oct. 1993.
- [2] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, and J. Peha, "Streaming video over the Internet: approaches and directions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 282–300, Mar. 2001.
- [3] P. Chou, A. Mohr, A. Wang, and S. Mehrotra, "Error control for receiver-driven layered multicast of audio and video," *IEEE Transactions on Multimedia*, vol. 3, no. 1, pp. 108–122, Mar. 2001.
- [4] W.-T. Tan and A. Zakhor, "Video multicast using layered FEC and scalable compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 373–86, Mar. 2001.
- [5] X. Li, M. Ammar, and S. Paul, "Video multicast over the Internet," *IEEE Network*, vol. 13, no. 2, pp. 46–60, Mar.-Apr. 1999.
- [6] C. Papadopoulos, G. Parulkar, and G. Varghese, "An error control scheme for large-scale multicast applications," in *Proceedings of IEEE INFOCOM'98*, vol. 3, 1998, pp. 1188–1196.
- [7] B. Girod and N. Farber, "Feedback-based error control for mobile video transmission," *Proceedings of the IEEE*, vol. 87, no. 10, pp. 1707–23, Oct. 1999.
- [8] Y. Ishibashi, Y. Tachibana, and S. Tasaka, "Responsiveness of layered multicast and feedback control for video traffic in the Internet," in *Proceedings of IEEE ICC'00*, vol. 2, 2000, pp. 846–52.
- [9] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," *IEEE Transactions on Networking*, vol. 5, no. 6, pp. 784–803, Dec. 1997.
- [10] J. Nonnenmacher, L. Martin, J. Matthias, E. Biersack, and G. Carle, "How bad is reliable multicast without local recovery," in *Proceedings of IEEE INFOCOM'98*, vol. 3, Mar. 1998, pp. 972–979.
- [11] I. Rhee, S. R. Joshi, M. Lee, S. Muthukrishnan, and V. Ozdemir, "Layered multicast recovery," in *Proceedings of IEEE INFOCOM'00*, vol. 2, 2000, pp. 805–813.
- [12] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 301–317, Mar. 2001.