

Optimal Allocation of Packet-Level and Byte-Level FEC in Video Multicasting over Wired and Wireless Networks

T.-W. Angus Lee S.-H. Gary Chan
Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon
Hong Kong

Qian Zhang Wen-Wu Zhu Ya-Qin Zhang
Microsoft Research, China
5F, Beijing Sigma Center, No. 49, ZhiChun Road
Haidian District, Beijing 100080,
P.R. China

Abstract—Multicast is an efficient technique to deliver video content over a network. In this paper, we consider such a multicast system to serve both wireless and wireline users when there is error over the wired network and wireless hop. Since packets are likely to be dropped in the wired networks while bit errors are more likely over the wireless hop, a combination of both packet-level and byte-level FEC is required to recover these errors. Given the estimated error and bandwidth characteristics reported by end users, the server needs to optimally allocate the packet-level and byte-level FEC to achieve maximum video quality. We study two schemes pertaining to whether or not the wireless gateway is able to transcode the video packets from the wired network before forwarding it to the wireless users. We first develop a model to analyze the system; and then propose an efficient algorithm for the FEC computation. We finally compare the schemes in terms of the optimal parameters used in the FEC, and the video quality achieved.

Keywords — Video Multicasting, Optimal FEC Allocation, error recovery, wireless Internet

I. INTRODUCTION

Multicast is an efficient technique to deliver video to its end users. In such a system, the video files stored or captured in a server are multicast to its clients distributed in a network. Data is usually multicast in a “TCP-friendly” manner, by which we mean that the streaming rate is not more than the available bandwidth in the network [1], [2]. The end client may be wired or wireless.¹ In the case of wireless network, the base station is generally connected to a gateway. In its simplest form, the gateway simply forwards whatever packets it receives to the wireless client without any re-packetization or fragmentation. A more sophisticated gateway, on the other hand, can take into consideration of the wireless characteristics and repacketizes the packets it receives from the wired infrastructure, adds or removes some information in them (e.g., error redundancy codes) before forwarding them to the end clients. This technique is known as “transcoding”, as data packets are re-encoded in the process. The “transcoding” gateway is beneficial since the packet error characteristics are different in the wired and wireless networks. In wired networks (such as the Internet), packets are lost mainly due to congestion at the routers, while in the wireless hop, packets are often lost due to random *bit* error caused by fading or multipath effect [3].

To recover packet loss, feedback recovery or forward error correction code can be used [4]. In general, feedback recovery does not work very well over large scale networks with real-time guarantee. FEC such as the Reed-Solomon (RS) code, on the other hand, is more appropriate for real-time communications by introducing some redundancy [5]. It consists of arranging the data and redundancy bits in such a way that even if a partial fraction of the bits are received, the original data may still be recovered. An FEC scheme adapting to the network error conditions is very efficient to maintain video quality. In this paper, we will mainly concerned with FEC to achieve maximum video quality. FEC strategies would be different over the wireless and wired networks, due to their different error characteristics. In the wired network, some packet-level FEC should be used, in which redundant or

parity *packets* are added so that packet loss can be recovered [3], [4], on the other hand, byte-level FEC should be used over the wireless hop, in which redundant or parity *bytes* within a packet can be added to recover bit error [3].

We consider that the clients periodically feedback to the source its “estimated” end-to-end bandwidth, packet loss and bit error rate between itself and the source (How to estimate the end-to-end bandwidth and packet drop rates accurately is beyond the scope of this paper. Interested readers are referred to [6] and references therein). Given these feedbacks, it is therefore of particular interest to address the following issue: under the multicast environment and given the heterogeneous error and bandwidth characteristics of its end users, how should the packet-level and byte-level FEC be allocated for a single video stream in order to maximize the overall video quality, in both cases of simple and “transcoding” gateways? We primarily concern the optimization for a single stream in this paper, since there are already many rich issues to be considered pertaining to FEC allocation and transcoding gateways. Our approach and discussion here readily extend to and would be useful for the layered multicast case, in which the video is multicast via multiple layers, and each layer has different FEC capability.

Traditionally video quality is measured by distortion given by PSNR [7]. It has been widely observed that such PSNR is proportional to the video goodput defined by useful data bits per second received by the end clients after FEC, given that the residual packet error rate is below a certain low value ($\leq 3\%$) [8]. Therefore, maximize video quality in PSNR is equivalent to maximize the overall goodput of the system, subjected to a certain low loss constraint.

Our contributions in this paper are 3-fold: i) We have studied video multicast to wired and wireless users; ii) We have investigated optimal allocation of packet-level and byte-level FEC for video delivery; and iii) We have developed a model and presented an analysis of transcoding gateway at the wireless hop. Our results show that the scheme with transcoding gateways outperforms the scheme with no transcoding gateways by a small margin, for wide ranges of packet drop rate.

We briefly present some previous works as follows. In [1], [2], Zhang et al. discussed rate adaptation approaches for video delivery. It consists of bandwidth estimation and adapting the transmission rate of the videos. Multiple video streams are carefully controlled by a quadratic rate-distortion function in order to unicast it in a “TCP-friendly” manner. Our work differs from these by considering video streaming in a multicast environment. Also, we consider the application of FEC on the video stream and the optimal partition of the available bandwidth for video data and parity bits. Recovering dropped packet by retransmission and delaying the playout time at the client have been discussed in [9], [10]. However, these works are related to wired environment and have not considered the wireless media and the optimal allocation of FEC codes. Other recovery schemes pertaining to limited retransmission and FEC (the so-called hybrid ARQ-FEC scheme), have been discussed in [11], [12]. Research has also been done sending delayed version of parity across different multicast groups, and clients can subscribe to different groups, according to their loss profiles and desired level of protection [13]. However, all of them

This work was supported, in part, by the Hong Kong Telecom Institute of Information Technology (97/98.EG01), and Sino Software Research Institute at the Hong Kong University of Science and Technology (SSRI00/01.EG04).

¹In this paper, we distinguish a “user” from a “client” in that a user is the person requesting a video, while a “client” is the station/machine the user uses.

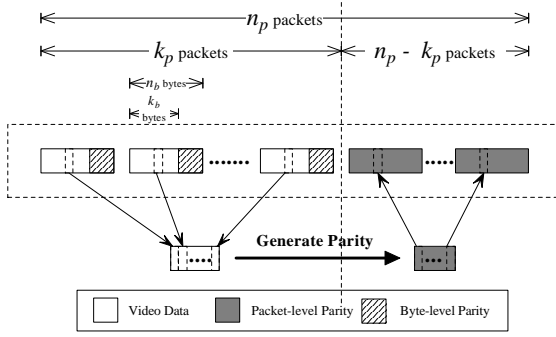


Fig. 1. Packet-level and byte-level FEC scheme for non-transcoding gateway.

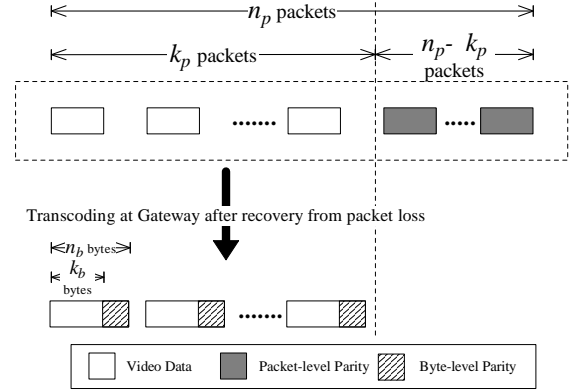


Fig. 2. Packet-level and byte-level FEC scheme for transcoding gateway.

are discussed under the context of a single medium only, and have not discussed “mixed” media (wired and wireless networks) and how packet-level and byte-level FEC can be optimally combined in such case. Furthermore, there has not been work examining the design of gateways for FEC.

This paper is organized as follows. We first present the schemes depending on whether the gateway transcodes the packets or not, and the goodput analysis in Sect. II. In Sect. III, we present some illustrative examples, and demonstrate the effectiveness of the schemes. We conclude in Sect. IV.

II. SCHEME DESCRIPTION AND ANALYSIS

Since every client has to receive the video with good quality, the bandwidth allocated to the video stream (including FEC encoding) should be equal to the minimum end-to-end bandwidth. Thus, the only concern here is how the error control should be applied to serve both wireless and wired clients so that its overall quality is optimized. As noted before, the quality is measured by the aggregate goodput in the system, or equivalently, average goodput of the client.

In this section, we first describe packet-level and byte-level FEC schemes in Sect. II-A. In Sect. II-B, we analyze and optimize video quality in terms of system goodput for non-transcoding and transcoding gateway, given client packet loss and bit error rate.

A. FEC Scheme Descriptions

We propose mixed packet-level and byte-level FEC to protect the video stream. We study the scheme with and without a transcoding gateway.

A.1 Non-transcoding gateway

With non-transcoding gateway, both packet-level and byte-level FEC encodings have to be done at the video server, and error correction are only done at the end clients. At the server, the compressed stream is first encoded with byte-level FEC followed by packet-level FEC. The decoding part is the reverse of the encoding process. Note that with this system, the byte-level FEC does not really help those wired clients (where packet drops occur) in improving their error resilience capability.

We show in Fig. 1 how to generate the two levels of FEC based on RS code. For the byte-level FEC, the encoder processes in symbols, where each symbol consists of $m = 8$ bits. Given a packet of size n_b bytes, $k_b (\geq 1)$ bytes of source data is packed with $n_b - k_b$ parity bytes, where $k_b = n_b, n_b - 2, \dots$. This is the so-called RS(n_b, k_b) code, which is able to correct up to t_b symbol errors in a packet, where $t_b = (n_b - k_b)/2$. The packet size n_b is limited by $2^m - 1$ symbols; therefore, for $m = 8$, $n_b \leq 255$.

With every k_p of these byte-encoded video packets, packet-level FEC is then applied to generate $n_p - k_p$ parity packets for a block of n_p packets, where $k_p = n_p, n_p - 1, \dots, 1$. This is generated as follows. The i th byte of each of the k_p video packets ($1 \leq i \leq n_b$) is taken out

to generate $n_b - k_b$ parity bytes. The generated parity bytes are then redistributed as the i th byte of each of the $n_p - k_p$ parity packets. Since all the packets are sequenced, up to $t_p = n_p - k_p$ packet losses in a block can be corrected. Clearly, as a block of packets has to be ready before packet-level FEC is done, the delay of the system increases with n_p . Therefore, in reality user delay requirement determines the n_p that can be used.

The server computes the optimal allocation between the video data rate, the packet-level FEC rate (i.e., the number of packet-level FEC parity bits per second) and the byte-level FEC rate (i.e., the number of byte-level FEC parity bits per second) given the feedbacks from the end clients. Let G be the multicast group size. The feedbacks for client g ($1 \leq g \leq G$) are in terms of the estimated end-to-end available bandwidth \hat{B}_g and the packet drop rate $\hat{P}_{l,g}$ ($\hat{P}_{l,g}$ may be estimated by the missing sequence numbers of the packets),² and, for wireless clients, the bit-error-rate of the wireless hop $\hat{e}_{b,g}$ ($\hat{e}_{b,g}$ may be estimated by using a two-state Markov process as given in [14]).

Given the feedback information, the server has to first decide the packet-level and byte-level FEC rates for the video stream, with its transmission rate including all the redundant bits is equal to the least end-to-end bandwidth in the multicast group (i.e., $R_0 = \min_g \hat{B}_g$). Let the packet-level FEC rate be R_p and byte-level FEC rate be R_b . Given (n_p, k_p) and (n_b, k_b) , R_p and R_b are clearly given by

$$R_p = R_0 \left(\frac{n_p - k_p}{n_p} \right), \quad (1)$$

and

$$R_b = R_0 \left(\frac{k_p}{n_p} \right) \left(\frac{n_b - k_b}{n_b} \right). \quad (2)$$

The video source rate R_s , defined as the data rate excluding all the FEC, is then given by $R_s = R_0 - R_p - R_b = R_0 \times (k_b/n_b) \times (k_p/n_p)$. The nomenclature used in this paper is listed in Table I.

A.2 Transcoding Gateway

A transcoding gateway transcodes video packets from packet-level FEC to byte-level FEC before forwarding the packets to the wireless clients. The gateway first recovers any dropped packet by the packet-level FEC, and then pads the video packets with byte-level FEC parity. Note that the wired clients need to perform packet-level FEC operations only, and, in contrast with the non-transcoding gateway, byte-level FEC encoding is done at the gateway rather than the server. We consider a simple transcoding gateway which does not do any

²We define a packet as “dropped” if the packet is in error during its transmission over the network. A dropped packet is (permanently) “lost” if it cannot be recovered after packet-level FEC.

TABLE I
NOMENCLATURE USED IN THIS PAPER.

G	: Size of multicast group (number of clients)
\hat{B}_g	: Estimated end-to-end available bandwidth for client g (bits/s)
$\hat{P}_{l,g}$: Estimated packet drop rate in the wired networks for client g
\bar{P}_l	: Average packet drop rate in the wired network
$\hat{e}_{b,g}$: Estimated bit error rate over the wireless hop for client g
\bar{e}_b	: Average bit error rate over the wireless hop
$e_{s,g}$: Symbol error rate in the wireless hop for client g
n_b, n_p	: Packet size of the byte-level FEC (bytes) and block size of the packet-level FEC (packets), respectively
k_b, k_p	: Data bytes in a byte-level FEC packet and number of packets in a packet-level FEC block, respectively
ε_o	: Constraint/Requirement on end-to-end packet loss rate (after error correction)
ϵ_g	: End-to-end packet loss rate (after error correction) for client g for non-transcoding gateway
η_g	: End-to-end packet loss rate (after error correction) for client g for transcoding gateway
Γ_g	: Goodput for client g (bits/s)
Γ	: Total goodput $\triangleq \sum \Gamma_g$ (bits/s)

packet fragmentation or reassembly. We see that a transcoding gateway achieves lower bandwidth requirement than the non-transcoding one (or equivalently higher video quality given a bandwidth constraint) by trading off some system complexity.

We show the detail of the encoding process in Fig. 2. The gateway first recovers the k_p data packets (out of the n_p FEC block) each of k_b bytes, and then transcodes the packets to n_b bytes by padding them with some byte-level FEC. Given the transmission rate of R_0 bits/s, the packet-level and byte-level FEC rates are clearly given by the same expressions as of Eqs. (1) and (2), respectively. The source rate is, however, given by $R_s = R_0 \times (k_p/n_p)$.

B. Quality Optimization

In this subsection, we analyze the system with non-transcoding and transcoding gateways and consider how the video quality can be maximized over all the clients in the system. As mentioned before, we consider minimizing the sum of PSNR over all the clients. For the error rate of interest, this is equivalent to maximizing the aggregate goodput Γ (bits/s), defined as the useful data bits delivered per second over all clients *after* error correction. Further let Γ_g be the goodput of the g th client.

Therefore, we study the following byte-level and packet-level FEC allocation problem: Given n_b and n_p , find the optimal n_p , k_p , and k_b in order to maximize

$$\Gamma = \sum_{g=1}^G \Gamma_g \quad (3)$$

such that the end-to-end packet loss rate (after error correction) is no more than a certain value ε_o (say, 0.01 – 0.03) over all clients. Here, we consider the sum of the individual goodput, i.e., all the clients in the system have the same priority or importance. If it is not that case, we need to assign some weight to each Γ_g (and thereof each PSNR). This extension is straightforward and would not be pursued further here.

B.1 Optimization for non-transcoding gateway

Let's consider a particular client g (and hence the subscript “ g ” in some of our equations) and obtain its goodput given $\hat{P}_{l,g}$ and $\hat{e}_{b,g}$. In the wireless hop, a symbol is considered in “error” if any of the m bits in the symbol are transmitted in error. Clearly, given bit error rate $\hat{e}_{b,g}$ in the wireless channel, the symbol error rate is $e_{s,g} = 1 - (1 - \hat{e}_{b,g})^m$. Since the RS(n_b, k_b) code corrects up to t_b symbol errors, the probability that a random packet cannot be recovered by byte-level FEC is given by

$$\alpha_g = \sum_{j=t_b+1}^{n_b} \binom{n_b}{j} e_{s,g}^j (1 - e_{s,g})^{n_b-j}. \quad (4)$$

Note that for the wired clients, $\alpha_g = 0$ as $\hat{e}_{b,g} = 0$ by definition.

A packet is “dropped” if it is dropped in the wired networks (with rate $\hat{P}_{l,g}$), or if it is in unrecoverable error (with probability α_g) over the wireless hop. Since the two events are independent, the end-to-end packet drop rate from the source to the client is given by $\beta_g = 1 - (1 - \hat{P}_{l,g}) \times (1 - \alpha_g)$.

Note that the dropped packets may be recovered by the packet-level FEC (see Fig. 1). Since up to $t_p = n_p - k_p$ dropped packets in the same block can be recovered by packet-level FEC. By considering the number of packet drops in an FEC block, the probability that a random packet is permanently “lost” (i.e., the end-to-end packet loss rate after error correction) is given by

$$\epsilon_g = \sum_{k=t_p+1}^{n_p} \frac{k}{n_p} \binom{n_p}{k} \beta_g^k (1 - \beta_g)^{n_p-k}. \quad (5)$$

The goodput of the client g is hence given by

$$\Gamma_g = R_0 \left(\frac{k_b}{n_b} \right) \left(\frac{k_p}{n_p} \right) \times (1 - \epsilon_g). \quad (6)$$

The allocation problem is a two dimensional search on k_p and k_b , which is of complexity $O(Gn_p n_b)$ and is not efficient. Validated by extensive runs, we found that packet-level FEC optimization can be done independently with that of byte-level FEC without affecting the results much (less than 1%). Therefore, we can greatly reduce the complexity to $O(G(n_p + n_b))$ by means of the following two-step procedure:

1. (*Packet-level FEC optimization*) For all clients, let $P_l = \max_g \hat{P}_{l,g}$. If $P_l \leq \varepsilon_o$, STOP and proceed to the next step (The packet drop rate is so low that $k_p^* = n_p$). Otherwise, for all the clients with $\hat{P}_{l,g} > \varepsilon_o$, search for the largest $k_p < n_p$ (for minimum overheads) such that ϵ_g (in accordance to Eq. (5)) of all these clients are no more than ε_o . This is the k_p^* required.
2. (*Byte-level FEC optimization*) Given k_p^* , find the largest $k_b < n_b$ such that ϵ_g for all the wireless clients are no more than ε_o . This is the k_b^* required.

B.2 Optimization for transcoding gateway

Consider a client g . The probability that a random packet is permanently lost over the wired network is clearly given by

$$\gamma_g = \sum_{k=t_p+1}^{n_p} \frac{k}{n_p} \binom{n_p}{k} \hat{P}_{l,g}^k (1 - \hat{P}_{l,g})^{n_p-k}. \quad (7)$$

If it is a wireless client, the packets corrected after packet-level FEC are transmitted over the wireless hop. The probability that these packets cannot be recovered due to wireless error has already been obtained

TABLE II
CLIENTS' PROFILE USED FOR TRANSCODING AND NON-TRANSCODING GATEWAYS.

	$\hat{P}_{l,g}$ (%)	$\hat{e}_{b,g}$ (10^{-4})
client 1	2.0572	0.9993
client 2	1.7179	0.5460
client 3	2.4790	0.8594
client 4	1.8248	1.3363
client 5	2.7698	1.0134
client 6	1.3341	0.0
client 7	2.1079	0.0
client 8	2.7578	0.0
client 9	1.1049	0.0
client 10	2.4529	0.0

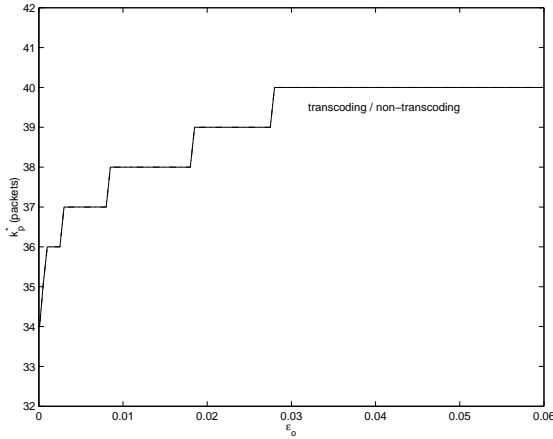


Fig. 3. k_p^* versus ϵ_o given k_b^* for transcoding and non-transcoding gateways ($\bar{e}_b = 10^{-4}$, $\bar{P}_l = 2\%$, $n_b = 255$ bytes, $n_p = 40$ packets).

as α_g in Eq. (4) (again, for the wired user, $\alpha_g = 0$). Therefore, the end-to-end packet loss rate after error correction is given by (by the independence of error rates in the wired and wireless networks)

$$\eta_g = 1 - (1 - \gamma_g)(1 - \alpha_g), \quad (8)$$

and hence the goodput of the client is

$$\Gamma_g = R_0 \left(\frac{k_p}{n_p} \right) \times (1 - \eta_g). \quad (9)$$

As in the non-transcoding case, we again observe that the packet-level FEC can be done independently of the byte-level FEC for the transcoding case. The optimization procedure is hence the same as that of the non-transcoding case, except that ϵ_g is replaced by η_g in Eq. (8).

III. ILLUSTRATIVE NUMERICAL EXAMPLES AND RESULTS

In this section, we compare the performance of transcoding and non-transcoding gateways. We consider a baseline system of $G = 10$ clients, with half of them wireless clients. We show in Table II $\hat{P}_{l,g}$ and $\hat{e}_{b,g}$ of each client, which are generated by assuming that they are uniformly distributed with mean $\bar{P}_l = 2\%$ and $\bar{e}_b = 10^{-4}$, respectively. Note that clients 1 to 5 are wireless clients, while the remaining are wired. The other baseline parameters are $\epsilon_o = 1\%$, $n_b = 255$, $n_p = 40$, and $R_0 = 100$ kbits/s (kb/s). Optimal FEC allocation will first be performed given these parameters. Then in our sensitivity analysis, we vary the other parameters, one at a time.

We show in Fig. 3 k_p^* versus ϵ_o for the transcoding and non-transcoding cases. Clearly, both cases have the same optimal k_p^* (due

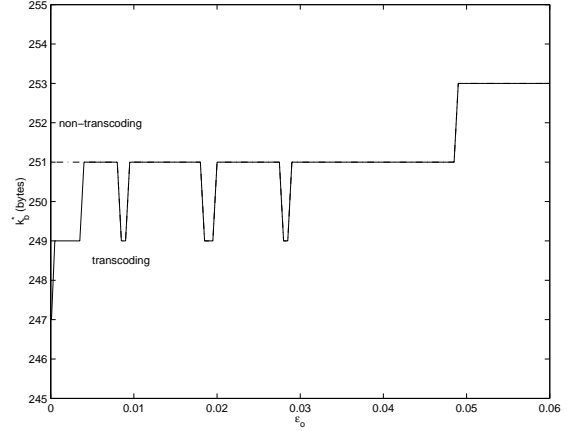


Fig. 4. k_b^* versus ϵ_o given k_p^* for transcoding and non-transcoding gateways ($\bar{e}_b = 10^{-4}$, $\bar{P}_l = 2\%$, $n_b = 255$ bytes, $n_p = 40$ packets).

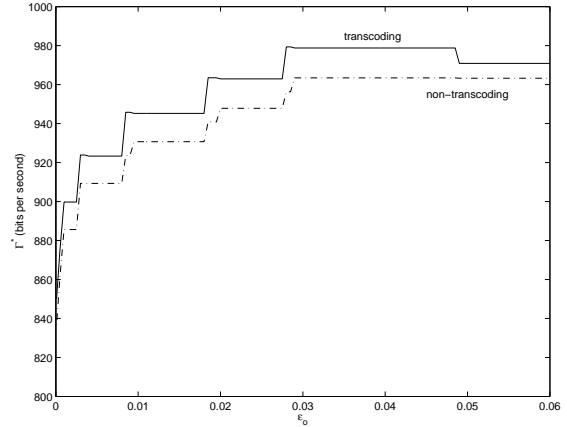


Fig. 5. Γ^* versus ϵ_o for transcoding and non-transcoding gateways ($\bar{e}_b = 10^{-4}$, $\bar{P}_l = 2\%$, $n_b = 255$ bytes, $n_p = 40$ packets).

to the same optimization step in k_p^*). k_p^* increases with ϵ_o in a step-wise manner (due to the constraint on integral value). Note that k_p is already very close to n_p , indicating that little packet-level FEC is necessary to achieve a low end-to-end packet loss. The packet-level FEC is so effective that even though most of the $\hat{P}_{l,g}$'s are greater than 2%, only a few overhead packets (2 in this case) are needed to bring ϵ_o to as low as 1%. No overhead is necessary when $\epsilon_o \geq \max_g \hat{P}_{l,g}$ (as all clients has $\hat{P}_{l,g} < \epsilon_o$).

We next show in Fig. 4 the corresponding k_b^* versus ϵ_o for transcoding and non-transcoding gateways. Both cases share almost the same k_b^* . As compared with k_p^* , k_b^* is quite insensitive to ϵ_o ; it increases relatively very slowly. Therefore, as ϵ_o changes, k_p is a more important parameter to adjust. Note that for $\hat{e}_{b,g} = 10^{-4}$, a random packet without any byte-level FEC is in error occurs with probability of $1 - (1 - \hat{e}_{b,g})^{mnb} = 0.18$. Even with this packet loss rate, only a few overhead bytes (about 4–6 in our plot) is enough to bring this error rate down to a low level given by ϵ_o . This again indicates the efficiency of byte-level FEC. The “dips” in the figure corresponds to the “rises” in Fig. 3. This is because once k_p^* is increased, the packet-level error correction capability decreases and hence a lower k_b^* (and thereof a stronger byte-level correction capability) is needed to compensate. As ϵ_o increases, the k_b^* jumps back up as the system can tolerate more end-to-end packet loss.

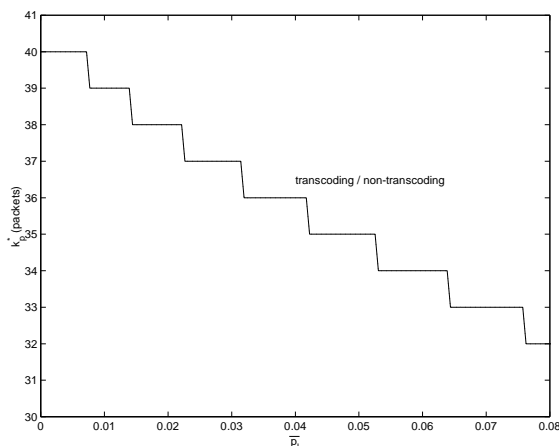


Fig. 6. k_p^* versus \bar{P}_l for transcoding and non-transcoding gateways ($\bar{\epsilon}_b = 10^{-4}$, $\epsilon_o = 1\%$, $n_b = 255$ bytes, $n_p = 40$ packets).

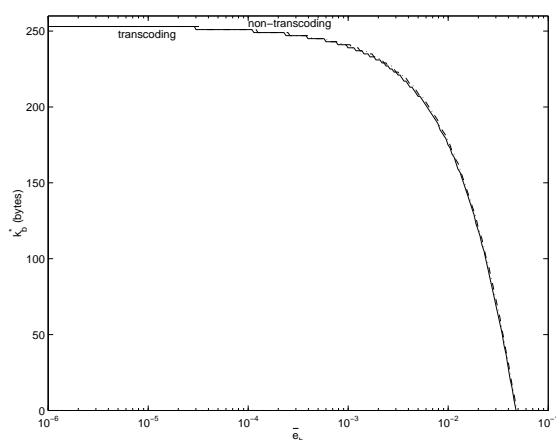


Fig. 7. k_b^* versus $\bar{\epsilon}_b$ for transcoding and non-transcoding gateways ($\bar{P}_l = 2\%$, $\epsilon_o = 1\%$, $n_b = 255$ bytes, $n_p = 40$ packets).

We show in Fig. 5 the corresponding optimal goodput Γ^* (i.e., with k_p^* and k_b^*) versus ϵ_o for the transcoding and non-transcoding cases. Though the goodput for the transcoding case is higher, there is no much difference between them (only about 2% difference here). This is expected because, from Eqs. (9) and (6), the ratio of the non-transcoding and transcoding goodputs for client g is given by $(k_b^*(\text{non-transcoding})/n_b) \times (1 - \epsilon_g)/(1 - \eta_g) \approx k_b^*/n_b$. From Fig. 4 we have already seen that k_b^* for the non-transcoding gateway is very close to n_b , and hence the difference is small. As ϵ_o increases, Γ^* in general first increases and then decreases (the decrease is shown for the transcoding case). This is due to the following. Γ is affected by two factors: i) the end-to-end packet loss rate ϵ_g and η_g (Γ decreases with them), and ii) k_p^* and k_b^* (Γ increases with them). From the figure, we see that when ϵ_o is small, the effect of k_p^* and k_b^* dominate, while when ϵ_o is higher, the error rate dominates and Γ^* decreases. For the cases of interest (i.e., $\epsilon_o \leq 5\%$), Γ^* increases with ϵ_o .

In Fig. 6, we show how k_p^* varies with \bar{P}_l . Clearly, k_p^* decreases with \bar{P}_l as more error protection is necessary. The packet-level FEC is quite effective, as only a few parity packets (given by $n_p - k_p^*$) are able to bring a high \bar{P}_l (say, 7–8%) to a low ϵ_o value (1%).

In Fig. 7, we show k_b^* versus $\bar{\epsilon}_b$ for transcoding and non-transcoding cases. Clearly, there is no much difference between these cases. In general, k_b^* decreases with $\bar{\epsilon}_b$ because more parity bytes are needed.

As $\bar{\epsilon}_b$ increases, k_b^* remains quite flat at the beginning and then sharply decreases. This indicates that when the bit error is high, many more parity bytes are needed to achieve a certain error rate after FEC. From the figure, we also see that when $\bar{\epsilon}_b$ is greater than a certain value (about 5% in this case), the bit error rate is too high that byte level FEC is no longer effective to bring errors in the wireless hop down to ϵ_o . Thus, the system can only tolerate wireless channel with bit error rate from 10^{-6} to 10^{-2} .

IV. CONCLUSIONS

In this paper, we have studied a video multicast system over wired and wireless networks with client feedback. The main challenge is to optimize the overall video quality by means of layer FEC allocations for the set of clients given their heterogeneous bandwidth and error characteristics, subject to a certain overall loss rate requirement. Furthermore, since there may be a transcoding gateway (which transcodes from packet-level FEC to byte-level FEC) between the wireless clients and the wired network, we have studied the value of such a gateway. We have analyzed the system and proposed an efficient allocation policy.

In order to serve all the clients, the transmission rate of the video stream should be equal to the minimum bandwidth of the clients. The issue of the video transmission is hence how to allocate packet-level and byte-level FEC so as to maximize video quality (in terms of goodput). Instead of a 2-dimensional search, we have presented an efficient algorithm for such optimal FEC allocation.

Our results show that the transcoding scheme performs only slightly better in terms of system goodput than the non-transcoding scheme (by about 2%). This is mainly due to the efficiency of FEC encoding (which occupies less than 20% of the data for the packet-level FEC, and less than 10% for byte-level FEC). This small difference may not justify the complexity of such a transcoding gateway. A gateway which transcodes data in some other ways may be more useful.

The schemes can be extended to apply in delivering layered video, which each layer is multicast in separate groups with different loss constraint. Future work can be done in combining the optimization in both FEC and bandwidth allocation across the layers.

REFERENCES

- [1] Q. Zhang, Y.-Q. Zhang, and W. Zhu, "Resource allocation for audio and video streaming over the Internet," to appear in special issue on Multimedia over IP in *IEEE Trans. on Multimedia*, September 2001.
- [2] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "Network-adaptive rate control and unequal loss protection with TCP-friendly protocol for scalable video over Internet," to appear in special issue selected from *IEEE ICME'00 on Multimedia Communications Journal of VLSI Signal Processing - System for Signal, Image and Video Technology*, 2001.
- [3] J. G. Kim and M. M. Krunz, "Bandwidth allocation in wireless networks with guaranteed packet-loss performance," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 337 – 349, June 2000.
- [4] K. Stuhlmüller, N. Farber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1012 – 1032, June 2000.
- [5] S. Wicker and V. Bhargava, *Reed-Solomon Codes and Their Applications*. IEEE Press, 1994.
- [6] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proceedings of ACM SIGCOMM'98*, pp. 303 – 314, October 1998.
- [7] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, vol. 15, pp. 23 – 50, November 1998.
- [8] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: a review," *Proceedings of the IEEE*, vol. 86, pp. 974 – 997, May 1998.
- [9] G. Ramamurthy and D. Raychaudhuri, "Performance of packet video with combined error recovery and concealment," in *Proceedings of INFOCOM '95*, pp. 753 – 761, April 1995.
- [10] A. Albanese, J. Blomer, J. Edmonds, and M. L. nd M. Sudan, "Priority encoding transmission," *IEEE Transactions on Information Theory*, vol. 42, pp. 1737 – 1744, November 1996.
- [11] R. Puri, K. Ramchandran, and A. Ortega, "Joint source channel coding with hybrid FEC/ARQ for buffer constrained video transmission," in *Proceedings of IEEE Second Workshop on Multimedia Signal Processing*, pp. 567 – 572, 1998.
- [12] S. S. Wang, H. Zheng, and J. Copeland, "A QoS enhanced hybrid SR-ARQ for mobile video communications," in *Proceedings of IEEE International Conference on Communications*, pp. 526 – 530, 2000.
- [13] P. Chou, A. Mohr, A. Wang, and S. Mehrotra, "FEC and pseudo-ARQ for receiver-driven layered multicast of audio and video," in *Proceedings of Data Compression Conference*, pp. 440 – 449, 2000.
- [14] H. S. Wang and P.-C. Chang, "On verifying the first-order markovian assumption for a rayleigh fading channel model," *IEEE Transactions on Vehicular Technology*, vol. 45, pp. 353 – 357, May 1996.