Distributed Resource Allocation for Network-Supported FGS Video Streaming

Antonios Argyriou
Philips Research, Eindhoven, The Netherlands

***Abstract*—In this paper, we propose a novel distributed resource allocation algorithm for fine granularity scalable video streaming. Our system model includes a proxy-based network infrastructure that is used for allocating locally the available bandwidth between the video rate and error control. Initially, we express analytically the expected decoder distortion as a function of both the source video streaming rate and the employed retransmission-based error control algorithm at each proxy. Next, we show that for this system configuration, the minimization of the decoder distortion is equivalent to a network flow control problem. We propose a distributed algorithm for solving this problem by applying Lagrange duality. The algorithm adapts the source streaming rate at the sender, while the number of retransmissions is optimized at each proxy individually. Simulation results demonstrate the effectiveness of the proposed scheme.**

I. Introduction

Due to the lack of QoS over the Internet, unicast media streaming systems have to monitor closely variations in the available bandwidth and the packet loss rate [1], [2]. A media sender can employ either rate control protocols or end-to-end measurements in order to estimate these parameters [3]. Accurate estimation is crucial for maximizing the system throughput and ensuring the optimal operation of other sophisticated adaptive media streaming algorithms (e.g. scheduling, error control, error resilient source coding). However, achieving this goal is difficult since a network path between two Internet hosts, may consist of several interconnected networks where each one is characterized by different bandwidth, delay, and packet loss characteristics. This means that end-to-end protocols and algorithms cannot distinguish the heterogeneity across the involved physical networks.

To avoid severe performance degradation, high discrepancies between interconnected networks are usually addressed with the strategic placement of proxies in the network [2], [4]. This approach has been studied extensively, especially for bridging wireless networks with the Internet [5], [6], [7]. Recently, several more sophisticated methods that are based on a proxy-based architecture, have also been reported. For example in [8], the authors present a methodology for distortion-optimized packetized media scheduling and streaming through a single intermediate proxy. The proposed system is based on monitoring the outcome of individual packet transmissions, based on which further distortion-optimized scheduling decisions are made. Clearly, this approach is computationally expensive when many proxies are involved.

This work was performed before the author joined Philips Research.

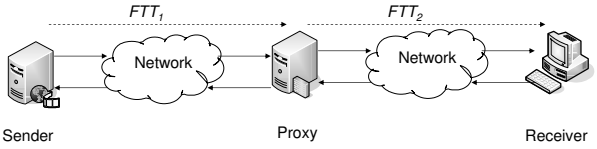

Fig. 1. System architecture for scalable video streaming with error control proxies. Each proxy measures the forward trip time (FTT) and the packet erasure rate ε.

Another interesting work that considers multiple proxies can be found in [10]. In that paper, the authors proposed a scheme that employs forward error correction (FEC) at intermediate nodes in the network. The disadvantages of that approach is the constant overhead required by FEC, and the uncoordinated, and therefore sub-optimal, selection of the FEC parameters.

Despite the wealth of research on network-supported media streaming, most of the existing works lack a generic modeling framework that considers the potential use of several media-aware network elements. In this paper, we take one step towards this direction, by adopting a system model that considers media-aware network nodes that employ automatic repeat request (ARQ) for recovering packets of a continuous media stream. Even though this functionality is fairly simple, it addresses one of the most fundamental problems in video communications, that is error control. The goal is to improve robustness in the face of network asymmetries, by allowing each proxy to optimally allocate the available bandwidth in a local scope.

Fig. 1 depicts a simplified view of our system model with a single proxy. We assume that the sender employs a fine granularity scalable (FGS) encoding scheme, since this approach allows faster adaptation of the source rate. The sender encodes the video into a base layer (BL) and an enhancement layer (EL). Furthermore, each video layer is protected with the use of unequal error protection (UEP) by adjusting the maximum number of allowed retransmissions. FGS data that belong to a specific layer and frame type, are packetized into the same network transport packet. This allows each proxy to exercise UEP easily since each network packet corresponds to particular layer. We use the notation $\vec{r}_h = \{r_h(l_0), r_h(l_1), ...\}$ to denote that at proxy h, the maximum retransmissions for packets of layer l_0 (base layer) is $r_h(l_0)$, for the first enhancement layer is $r_h(l_1)$, etc. The sender and each proxy execute a resource allocation algorithm that calculates

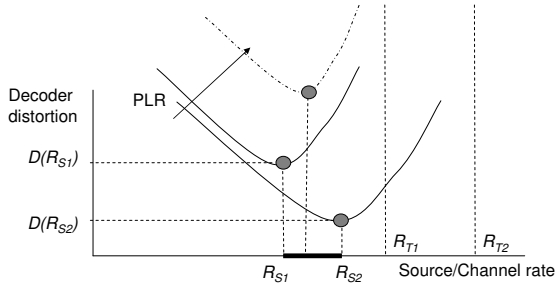

Fig. 2. Convex hull of the rate-distortion function for FGS video transmission under different channel realizations.

both the globally optimal source streaming rate, and the optimal rate for error control. In order for the sender and each proxy to be able to derive the optimal allocation, the expected distortion at the decoder is expressed by considering the effect of the ARQ algorithm implemented at each proxy.

A. Paper Organization

The rest of this paper is organized as follows. In section II, we highlight the main idea behind the proposed distributed resource allocation and error control scheme. Next, in section III we describe the functionality implemented at each proxy and we also calculate its effect on the overall packet loss rate. Subsequently, in section IV we present the source and channel distortion models for FGS encoded video. We formulate the problem of joint resource allocation and distributed error control in section V. Section VI presents our comprehensive simulation results while in section VII we conclude this paper.

II. Video Transmission with Multi-Hop Error Control

In order to demonstrate more clearly the main motivation behind the system we described, we will examine its theoretical rate-distortion (RD) performance. Figure 2 presents the convex hull of the RD function for fine-granularity scalable (FGS) video transmission over packet erasure channels. The y axis shows the sum of both source and channel distortion [11], while the x axis depicts the source/channel rate allocation. Now, the two lower curves correspond to an encoded bitstream transmitted over two different channel realizations. When each of these channels is considered individually, the optimal source streaming rates are denoted as R_{S1} and R_{S2}, while the optimal rates for error control are given as $R_{T1}-R_{S1}$ and $R_{T2}-R_{S2}$ respectively. However, when the two channels are connected in tandem, the convex hull of the joint RD function is different which means that the optimal source/channel rate allocation is also different. In the later case, if we employ an end-to-end mechanism for probing the end-to-end path (with a rate control protocol), the bottleneck link would determine the "optimal" source rate. The residual packet erasure rate measured at the source, will determine the optimal rate that should be allocated for error control. The remaining bandwidth, that is not probed at the second path, can be used for error control at the discretion of the proxy that resides between the two communication channels. However, even this solution is sub-optimal as it is demonstrated with this figure, since the sender would select R_{S1} as the optimal streaming rate since it does not know the existence of the extra bandwidth $R_{S2}-R_{S1}$. Our goal is to devise an algorithm that can identify the actual minimum operating point for the joint RD function. This can only be done if the proxy has more active role during the streaming session. In the next sections, we demonstrate how this idea can be realized in a practical algorithm.

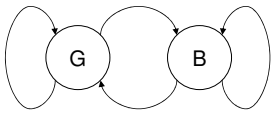

Fig. 3. Two-state Markov chain channel model.

III. The Error Control Proxy

According to our system model, error control is exercised with ARQ between two successive proxies. The advantage of ARQ is the simple implementation, low processing overhead, and the fact that no bandwidth cost is incurred when no packet losses take place. For the rest of this section, we discuss in detail the functionality that is implemented at each proxy, its effect on the packet error rate experienced by video packets.

A. Channel Model

The communication channel between two proxies is packet-based, and the adopted packet loss model is that of an "erasure channel". We adopt a Markov chain for characterizing the transitions from good and bad states for each individual channel (Fig. 3). For this channel model, if the network layer packet erasure rate for proxy/channel h is denoted as ε_h and $m_h(l)$ is the average number of retransmissions for packets of video layer l, the resulting residual packet loss rate is given by $\varepsilon_h^{m_h(l)}$.

Even though this channel model assures independent packet erasure probabilities between interconnected proxies, it does not completely de-correlate the overall end-to-end packet loss rate for delay-sensitive multimedia streaming. In our system model, de-correlation is achieved by certain design choices, that we explain in the next subsection.

B. ARQ Performance Model

First, each flow terminates at each proxy which means that only packets that have been successfully received at proxy h, will be considered as part of a new flow until proxy $h+1$. This concept is implemented with the real-time transport protocol (RTP) [12]. Furthermore, each proxy measures the packet erasure rate ε_h for the

channel that is attached, based on the real-time control protocol (RTCP) receiver reports [12]. This allows each proxy to estimate separately the effect of error protection decisions on the quality of the transmitted video. However, it is possible that the decoder receives a continuous RTP stream but there is a gap in the corresponding media payloads between two successive RTP packets. The ARQ at the proxy is responsible for ensuring that the application payload that is missing, corresponds to a media packet that has low importance for the decoding of the transmitted video sequence. This is achieved by controlling the retry limit for packets of a particular layer.

Besides ε_h, both the forward trip time ($\widehat{FTT}_h$) and the round-trip time ($\widehat{RTT}_h$) are also measured between two proxies. This allows each proxy to identify packets that were not lost by an erasure, but have violated the maximum allowed one-way delay τ_h that is specific for each proxy/channel h. By selecting the value of τ_h to be proportional to the measured $\widehat{FTT}_h$, the proxy can estimate the appropriate number of retransmissions that should exercise. This is done by calculating $FTT_h(l)$ which is the forward trip delay for the corresponding channel when ARQ is also considered.

Based on our previous description of the proxy functionality, the overall packet loss rate experienced by packets of layer l is:

$$\rho_h(l) = \varepsilon_h^{m_h(l)} + (1-\varepsilon_h^{m_h(l)}) P_r\{FTT_h(l) > \tau_h\} \quad (1)$$

Our goal now is to express analytically the term $P_r\{FTT_h(l) > \tau_h\}$ in the previous equation, since it expresses the probability that packets are late for their prescribed deadline.

An important parameter that must be estimated first, is the distribution of the actual number of used retransmissions for a given allowable range. This value is easily calculated according to the adopted Markov channel model. The probability that i retransmissions are needed until a successful transmission is given by $(1-\varepsilon_h)\varepsilon_h^i$. On the other hand, the probability that a packet needs i retransmissions out of the $r_h(l)$ allowable for packets of layer l is given by:

$$\pi(i, r_h(l)) = \frac{(1-\varepsilon_h)\varepsilon_h^i}{1-\varepsilon_h^{r_h(l)+1}} \quad (2)$$

Therefore, the average source packet transmission delay for each proxy can be written as follows:

$$FTT_h(l) = \sum_{i=0}^{r_h(l)} \pi(i, r_h(l))[i * \widehat{RTT}_h + \widehat{FTT}_h] \quad (3)$$

Also, given $r_h(l)$ and ε_h, we can calculate the average number of retransmissions per packet for proxy h as:

$$m_h(l) = \frac{1-\varepsilon_h^{r_h(l)+1}}{1-\varepsilon_h} \quad (4)$$

The previously derived equations, are used by the proxy in order to regulate the optimal level of unequal error protection for each enhancement layer of the FGS video stream. This is done by calculating the optimal retry limit vector $\vec{r}_h$ so that the introduced channel distortion is minimized. In the next section, we will derive a model for the expected end-to-end video distortion, while in a later section we will demonstrate how it can be used by each proxy.

IV. Rate-Distortion Model for FGS Encoded Video

In this paper, we assume the use of the Scalable Video Codec (SVC) fine-granularity scalable (FGS) encoding scheme [13]. With FGS encoding, the input video sequence is compressed in a base layer (BL), that contains the most important information, while many enhancement layers (EL) can be encoded. The base layer consists of a standard single-layer H.264 bitstream while the enhancement layer is coded with the bitplane technique and consists of the so-called progressive refinement slices [14]. The advantage of using FGS encoded video for real-time streaming, is the ability to adapt to bandwidth fluctuations by truncating the EL at arbitrary points. Practically, this means that the quality improvement gained from the enhancement layer, is proportional to the number of received enhancement layer bits [17]. Even though the FGS enhancement layer can be considered as a single layer that provides continuous RD operating points, we develop our framework in terms of discrete layers. This approach will allow easier formulation of practical error protection strategies [15], [16].

Furthermore, we consider two types of distortion in our model: a) The source distortion (D_S) that is caused by the lossy video encoder, and b) the channel distortion (D_C) that is caused by the loss of video packets in the network. In the next two subsections, we calculate these two components.

A. Source Distortion

The distortion introduced at the source, depends of course on the rate allocated for source coding. The overall bitrate of scalable encoded bitstream can be expressed in terms of the base and enhancement layer source rates. For a total of N layers, we have:

$$R_s = \sum_{l=0}^{N-1} R_s(l), \quad (5)$$

where $l=0$ denotes the base layer, while an increasing l denotes enhancement layers of decreasing importance. The corresponding source distortion can be decomposed in the same way as before:

$$D_S = \sum_{l=0}^{N-1} D_S(l) \quad (6)$$

For expressing the relationship between rate and distortion of FGS encoded video, we adopt the statistical model reported at [17]. The derived rate-distortion function for FGS video was expressed in that work as follows:

$$D_S(R_S) = -2^{\alpha R_S + b\sqrt{R_S} + c} \quad (7)$$

The parameters α, b and c are evaluated empirically, since they depend on the video bitstream. These parameters are calculated offline during the encoding of the video sequence, and they remain constant during the real-time streaming [18].

B. Channel Distortion

For estimating the channel distortion, the effect of error control employed at each proxy must be calculated individually. Assume that the size of the EL data for a single video frame is K, while k_l is the data size of layer l (i.e. $K = \sum_{l=1}^{L} k_l$). Then the channel rate dedicated both for retransmissions at proxy h is:

$$R_{C_h} = \sum_{l=0}^{L} m_h(l) k_l \tag{8}$$

for a specific proxy h. The channel distortion introduced by the un-successful decoding of layer l, will be:

$$D_C(l) = D(k_l) - D(k_{l-1}) \tag{9}$$

The above quantity is easily calculated since it is proportional to the size of the data k_l and k_{l-1}. The probability that the channel distortion introduced at all layers for a frame transmitted through proxy h, will be:

$$D_{C_h} = \sum_{l=0}^{L} \rho_h(l) D_C(l) \sum_{i=0}^{l-1} \rho_h(i) \tag{10}$$

The last equation expresses the channel distortion introduced at each layer l, by considering the lower layers that are needed for the successful reconstruction of l.

V. Distributed Resource Allocation

Armed with an expression of source and channel distortion for a network with multiple error control proxies, we can proceed and define the global optimization problem. Intuitively, it can be seen that this is a network resource allocation problem, since the goal is to calculate the optimal resource allocation between source and error control at each particular proxy. Therefore we formulate it as a constrained optimization of a utility function [19]. In our case, the utility function that must be minimized is the decoder video distortion. Recall that $\vec{r}_h$ is the vector that contains the maximum number of retransmissions for media packets at proxy h. Then the optimization problem we try to solve, requires the minimization of the expected total video distortion D_t, subject to the channel rate constraint at each proxy:

$$\begin{aligned} &\min\ D_t(N, \vec{r}_1, \vec{r}_2, ..., \vec{r}_H) \\ &\text{subject to } R_h(N, \vec{r}_h) \leq R_{T_h} \quad \forall h \in H \end{aligned} \tag{11}$$

Both N and $\vec{r}_h$, are the optimization variables, while R_h is the total bitrate injected to channel h. Let us re-write (11) by expressing distortion as a function of each link's contribution to the global video distortion as follows:

$$\begin{aligned} &\min\ D_S + \sum_{i=1}^{H} D_{C_h} \\ &\text{subject to } R_S + R_{C_h} \leq R_{T_h}, \forall h \in H \\ &R_S \in \mathbf{S}, R_C \in \mathbf{C} \end{aligned} \tag{12}$$

This simple re-writing, reveals a very important issue regarding the nature of the problem. Equation (**??**) essentially means that this is is a flow control optimization problem since the goal is find the optimal source streaming rate R_S, and the optimal error control rate R_C, that minimize the utility function [21]. The derivation of the optimal of the optimal $\vec{r}_h$ is a task that is embedded in the minimization of D_{C_h}, and can be derived after the optimal rate for error control has been calculated.

Furthermore, according to (9), the channel distortion is directly proportional to the number of enhancement layer lost bits. This means that the channel distortion can be calculated locally at each proxy from (10), since it knows the size of the transmitted packets and the layer they belong. This is a key observation, since it means that the utility function is separable and a distributed solution to this problem can be derived via dual decomposition [19]. Therefore, we can apply Lagrange duality to the constraint in (12), and produce the partial Lagrangian as follows:

$$L = D_S + \lambda^T \vec{D}_C + \vec{\lambda}^T \left\{ \vec{R}_T - R_S - \vec{R}_C \right\} \tag{13}$$

In this equation $\vec{\lambda}$ is the vector of the Lagrange multipliers, while the size of the vector is equal to the number of proxies. The dual function is then defined as:

$$g(\lambda) = \max_{R_S \in \mathbf{S}, R_C \in \mathbf{C}} L(R_S, \vec{R}_C, \lambda) \tag{14}$$

Therefore, the dual problem is:

$$\begin{aligned} &\min\ g(\lambda) \\ &\text{subject to } \lambda \geq 0 \end{aligned} \tag{15}$$

In [21], it was proven that strong duality holds for this type of flow control problems. This means that the dual algorithm converges to an optimal solution that is also optimal for the primal problem defined in (12). Since the dual function of our problem is also separable, both the optimal source streaming rate at the sender, and the optimal error control rate at each proxy, can be optimized individually. The sender will solve

$$R_S^*, R_{C_1}^* = \arg \min_{R_S \in \mathbf{S}, R_{C_1} \in \mathbf{C}} \left\{ D_S + D_{C_1} - \lambda^T (R_S - R_{C_1}) \right\}, \tag{16}$$

while the rest of the proxies will solve:

$$R_{C_h}^* = \arg \min_{R_{C_h} \in \mathbf{C}} \left\{ D_{C_h} - \lambda^T (R_S - R_{C_h}) \right\} \tag{17}$$

For calculating the Lagrange multipliers, the following gradient method can be used at each proxy [20]:

$$\lambda_h^+ = \lambda_h - \beta \left(R_{T_h} - R_{C_h} - R_S \right) \tag{18}$$

In this equation β is the step length. The effectiveness of the resource allocation algorithm depends considerably on the timely dissemination of $\vec{\lambda}$ and the selection of β. We must note at this point that the sender does not have to notify each proxy about the optimal resource allocation since this information is implicitly communicated through the selected optimal source rate. This means that in our system, the sender calculates a source streaming rate that

will enable globally optimal resource allocation between source and error control for each of the channels in the end-to-end path.

Practically, the system operates as follows. Assume for example that the available channel rate of a particular proxy R_{T_h} increases. The corresponding proxy can take advantage of the extra bandwidth by increasing the retry limit for each layer in the error control vector $\vec{r}_h$. The Lagrange multiplier given in (18), is also reduced in order to reflect this change. To avoid under-utilization, this new value is communicated to all the proxies. The change of λ is reflected in the calculation of (16) and (17). First, the source rate starts to increase in order to maximize the last term in (16). For this increase in R_S, the other components in (16) are recalculated. The result is a new optimal source rate R_S^*.

VI. Performance Evaluation

In this section we investigate the performance of the resource allocation algorithm for different end-to-end proxy-based topologies. The CIF sequence Foreman with 300 frames was encoded at 30 fps with the latest JSVM software [13]. For achieving low delay we adopted the IBPBPB structure with integer motion vectors, wile the base layer was encoded at a constant bitrate of 128Kbps. The one-way forward and backward trip time are set equal to transmission time of three frames. The proposed system is named joint resource allocation with distributed error control (JRADEC), in order to indicate that resource allocation between the source rate and the error control rate is optimized jointly but in a distributed fashion. We also name a typical system as RADEC to indicate that source rate adaptation and error control are implemented without coordination at the sender and each proxy respectively. Our simulation results correspond to a constant random channel. Finally, we averaged the PSNR results for 20 realizations of the random channel.

A. Single Proxy

1) Constant erasure rate: Fig. 4 presents the RD performance for a network configuration with a single proxy. Both connected channels are configured with similar packet erasure rate equal to 2%. PSNR is presented in the *y* axis, while the *x* axis corresponds to the available channel rate on the first channel. In the lower part of the figure, we also present the optimal source rate calculated by the two algorithms. The first interesting observation that can be done based on this figure, is that when R_{T2} is low, for every value of R_{T1}, both systems exhibit minor differences in performance (at most 1dB). This happens because a low R_{T2} suppresses the upper bound of the source streaming rate. Indirectly it also allows most of available rate on the first channel to be used for error control and lead thus into negligible channel distortion. On the other hand, for increased values of R_{T2} and low R_{T1}, the situation in the two channels reversed, but the difference in PSNR is again marginal. The reason is the same since the second channel introduces negligible channel distortion because of the significant spare bandwidth

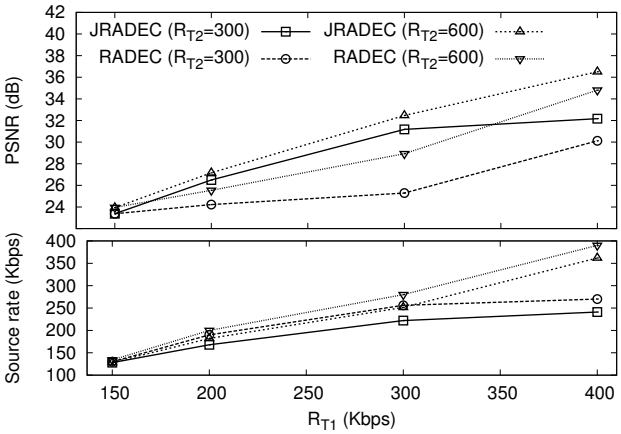

Fig. 4. PSNR results for streaming through a single error control proxy with constant erasure rates ε_1 and ε_2 equal to 2%.

available for error control. However, when the network does not operate in areas of highly asymmetric channel rates, our system offers considerable improvements in the decoded signal quality. The PSNR increase is close to 4-5dB for channel rates between 200-300Kbps. The reason why JRADEC outperforms the uncoordinated RADEC, is because the sender can calculate an optimal source rate from a wider range that is not limited by any of the two channels. In fact, we can see in the lower part of Fig. 4 that the optimal streaming rate calculated by JRADEC is always lower than RADEC. The proposed system is more aggressive in allocating part of the channel rate for error control since it leads to improvements in the overall quality. This last observation is critical and supports our claim that the globally optimal resource allocation between source and error control, can be unidentified if the problem is not considered in a joint optimization setting.

2) Constant channel rate: Fig. 5 presents results for a configuration where the packet erasure rate ε_2 varies between 1% and 5%, while the channel rates are set equal to 300 Kbps. Our observations start by noting that for low ε_2, the performance is the same regardless of the employed algorithm. The reason is that low ε_2 corresponds to few packet losses and so the contribution in the channel distortion is minimal. However, as ε_2 is increased, the contribution on the channel distortion is also increased. The RADEC scheme is impossible to account for this situation on the second channel, and as a result it fails to select a source rate that will enable optimal error control on the second proxy. A more closer observation reveals another interesting fact. When the discrepancies in the packet erasure rate are in the range of 50% or more, even with low absolute value, the differences in the resulting PSNR are significant. This means that it is more critical to use the proposed algorithm when for example ε_1 and ε_2 are equal to 1% and 2% respectively, than when they are both equal to 5%. Finally, we observe again that the results for the selected source rate, indicate

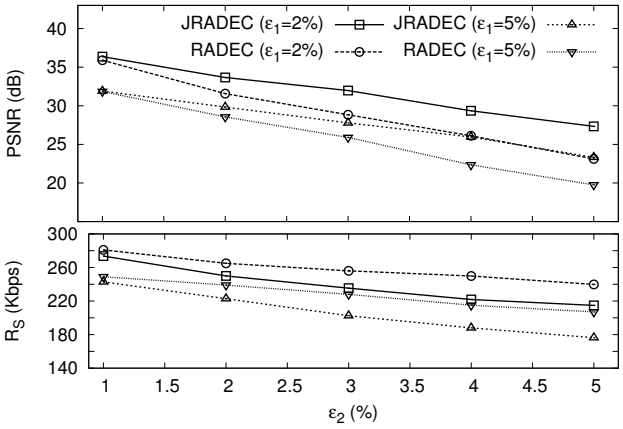

Fig. 5. PSNR results for streaming through a single error control proxy with constant channel rates R_{T1} and R_{T2} equal to 300Kbps.

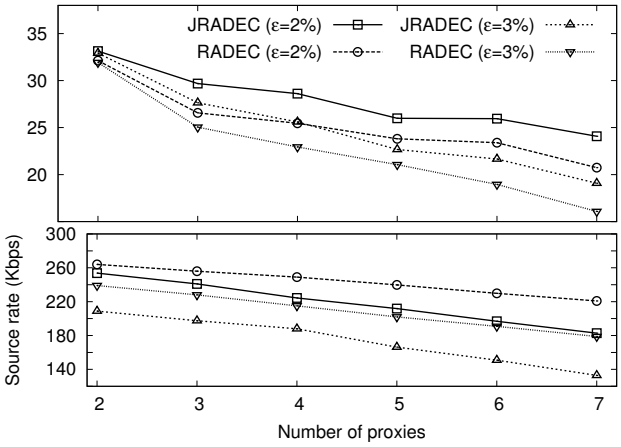

Fig. 6. PSNR results when multiple error control proxies are connected in tandem. The bandwidth of the corresponding channels alternate between 500 Kbps and 300 Kbps.

a more aggressive allocation towards error control from JRADEC as ε_2 is increased. This is something that should be expected given the need for more aggressive error control when packet losses and channel distortion are increased.

For the previous experiment, we present in table I the optimal configuration for the ARQ retry limit as it was calculated by the two algorithms under test. Note that these are just the first four layers of the bitstream and they do not correspond to the actual number that either system sent during each simulation. Nevertheless, these results are very useful for portraying at low-level the behavior of both systems. Overall, the proposed scheme is using increased portion of the available channel rate for retransmissions which explains the considerably higher retry limits. Note that this is not the actual number of retransmissions exercised. Furthermore, the relatively high values for the retry limit is because they are exercised between proxies and not end-to-end when they are needed. In the later case, the retry limit would be of course quite lower, since retransmissions would introduce significant delay.

TABLE I
ARQ RETRY LIMIT FOR DIFFERENT ERASURE RATES AND AND FOUR ENHANCEMENTS LAYERS OF THE FGS BISTREAM.

Algorithm	Erasure rate (%) $(\varepsilon_1,\varepsilon_2)$	Calculated retry limit (r_1,r_2)			
		l_1	l_2	l_3	l_4
JRADEC	(2,3)	(5,6)	(4,4)	(3,4)	(0,1)
	(2,5)	(4,7)	(3,6)	(2,4)	(0,2)
	(5,3)	(8,3)	(6,4)	(3,1)	(2,0)
	(5,5)	(8,8)	(6,6)	(4,2)	(2,1)
RADEC	(2,3)	(2,2)	(2,1)	(1,1)	(0,1)
	(2,5)	(2,3)	(2,1)	(1,3)	(0,2)
	(5,3)	(4,2)	(3,2)	(4,0)	(3,1)
	(5,5)	(5,3)	(3,2)	(1,1)	(0,0)

B. Multiple Proxies

After considering the very insightful but simple topology with a single proxy, in this subsection we present representative results for multiple proxies. For our simulations, we set the available rate on the channels with even index to be 500 Kbps, while the proxies with odd index correspond to a channel of 300 Kbps. The packet erasure rate is the same for all the channes involved in the same simulation, while two sets of simulations were performed for 2% and 3% respectively. PSNR results can be seen in Fig. 6.

As it is expected, the addition of more packet erasure channels on the end-to-end path decreases the PSNR due to increased packet loss. However, the JRADEC system is able to minimize the quality degradation, since the same basic principle we described before, for the case of a single proxy, also applies here. Furthermore, for the second set of simulations where the packet erasure rate is 3% for all channels, the PSNR gain is still significant. This essentially means that even for low packet erasure rates our algorithm is very useful for enhancing robustness and improving quality. Notice also that when a proxy with even index is added, the PSNR decrease is quite lower due to the higher available channel rate of 500 Kbps. Another important observation from these results is that for low number of packet erasure channels, the use of the proposed scheme outperforms a RADEC system that operates under slightly better channel conditions (i.e. $\varepsilon = 2\%$). This result is very important and can be explained if have a look into the lower part of the figure where the optimal source rates are presented. We see for example that for 2 and 3 proxies and $\varepsilon = 2\%$, RADEC calculates a very high source rate given the rapid increase in ε. This can also be explained by noting that with the addition of new proxies in the end-to-end path, the overall packet erasure rate is increased faster for the set of simulations with $\varepsilon = 3\%$.

VII. Conclusions

In this paper, we developed a distributed resource allocation and error control algorithm suitable for FGS video streaming over the Internet. Our system model leverages a proxy-based infrastructure for exercising error control with ARQ. The objective of the algorithm is to adapt the source streaming rate at the sender, and the number of retransmissions at each proxy in a coordinated and globally optimal fashion. The main advantage of our algorithm, is its distributed nature and the minimum exchange of control information between the involved proxies. Furthermore, our results show that it is able to maintain considerably higher PSNR at the decoder when compared with uncoordinated resource allocation schemes, since it achieves minimum quality degradation in asymmetric channel conditions.

References

[1] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*. Prentice Hall, 2002.

[2] M. van der Schaar and P. Chou, *Multimedia over IP and Wireless Networks*. Academic Press, 2007.

[3] M. Handley, S. Floyd, J. Pahdye, and J. Widmer, "TCP friendly rate control (TFRC): Protocol specification," RFC 3448, January 2003.

[4] L. Gao, Z.-L. Zhang, and D. Towsley, "Proxy-assisted techniques for delivering continuous multimedia streams," *IEEE/ACM Transactions on Networking*, vol. 11, no. 6, pp. 884–894, December 2003.

[5] T. C. Wang *et al.*, "Low-Delay and Error-Robust Wireless Video Transmission for Video Communications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 12, pp. 1049–1058, December 2002.

[6] A. C. Begen and Y. Altunbasak, "Videoconferencing over an intermediate-proxy," in *IEEE International Conference on Image Processing (ICIP)*, 2004.

[7] W. Tu and E. Steinbach, "Proxy-based error tracking for h.264 based real-time video transmission in mobile environments." [Online]. Available: citeseer.ist.psu.edu/tu04proxybased.html

[8] J. Chakareski and P. Chou, "Radio edge: Rate-distortion optimized proxy-driven streaming from the network edge," *IEEE/ACM Transactions on Networking*, vol. 14, no. 6, pp. 1302–1312, December 2006.

[9] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *Microsoft Research Technical Report MSR-TR-2001-35*, 2001.

[10] Y. Shan, I. Bajic, S. Kalyanaraman, and J. Woods, "Overlay multi-hop fec scheme for video streaming over peer-to-peer networks," in *International Conference on Image Processing (ICIP)*, 2004.

[11] G. Cheung and A. Zakhor, "Bit allocation for joint source/channel coding of scalable video," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 340–356, March 2000.

[12] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," RFC 3550, July 2003.

[13] H. S. J. Reichel and M. W. (eds.), "Joint scalable video model jsvm-6," Joint Video Team, Doc. JVT-S202, 2006.

[14] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable H.264/MPEG4-AVC extension," in *IEEE International Conference on Image Processing (ICIP)*, 2006.

[15] B. Lamparter, A. Albanese, M. Kalfane, and M. Luby, "PET - priority encoding transmission: A new, robust and efficient video broadcast technology (video)," in *ACM Multimedia*, 1995, pp. 547–548.

[16] Q. Zhang, W. Zhu, and Y. Zhang, "Channel-adaptive resource allocation for scalable video transmission over 3G wireless network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 8, pp. 1049–1063, August 2004.

[17] M. Dai and D. Loguinov, "Analysis of rate-distortion functions and congestion control in scalable internet video streaming," in *NOSSDAV*, 2003.

[18] J. Yan, K. Katrinis, M. May, and B. Plattner, "Media- and TCP-friendly congestion control for scalable video streams," *IEEE Transactions on Multimedia*, vol. 8, no. 2, April 2006.

[19] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE Transactions on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, August 2006.

[20] D. P. Bertsekas and R. G. Gallager, *Data Networks*. Prentice-Hall, 1987.

[21] S. Low and D. Lapsley, "Optimization flow control-i:Basic algorithm and convergence," *IEEE Transactions on Networking*, vol. 7, no. 6, pp. 861–874, 1999.