Distributed Packet Scheduling of Multiple Video Streams over Shared Communication Resources

Jacob Chakareski and Pascal Frossard
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Institute - LTS4, CH-1015 Lausanne, Switzerland

Abstract— **We consider the problem of distributed packet selection and scheduling for multiple video streams sharing a communication channel. An optimization framework is proposed to enable the multiple senders to coordinate their packet transmission schedules, such that the overall quality over the video clients is maximized. The framework relies on rate-distortion information that is used to characterize a video packet and that consists of two quantities: the size of the packet in bits, and its importance for the reconstruction quality of the corresponding stream. Using the framework, each of the senders allocates to its own video packets a share of the bandwidth available on the communication channel, that is proportional to the relative importance of these packets. Thereby, a decentralized streaming strategy is provided that allows for trading-off rate and distortion, not only within a single video stream, but also across different streams. Simulation results demonstrate that, for the difficult case of scheduling non-scalably encoded video streams, our framework substantially outperforms a conventional streaming system that does not consider the relative importance of the video packets. The gains in performance reach up to 8 dB in both streaming scenarios under examination, namely adaptation to random packet loss and simultaneous adaptation to packet loss and available bandwidth.**

I. INTRODUCTION

The demand for multimedia traffic sent over the Internet exhibits an ever growing trend today [1]. Therefore, scenarios where multiple media streams have to share common resources are becoming increasingly frequent. Transmission of concurrent media streams in a wireless LAN environment is one such example. In that context, it becomes important to consider the performance of the whole streaming system, in order to maximize the overall quality of service to all users.

Performing proper video packet selection and scheduling in such a setting can be an involved task. When a sender is allocated an insufficient transmission bandwidth, it will need to reduce its transmission rate in order to account for it. This in turn is achieved by omitting packets prior to transmission due to their timing constraints imposed by the underlying streaming application. Scalable coding techniques [2] have been developed to solve these problems, where the scalable encoding provides an inherent prioritization among the compressed data which in turn provides a natural method for selecting which portions of the compressed data to deliver, while meeting the transmission rate constraints. However, scalable streams have not gained a wide acceptance due to a few shortcomings, e.g., their coding inefficiency. On the other hand, non-scalable or non-prioritized video content, is predominantly used in streaming today, but it unfortunately does not suggest a natural method of placing delivery priorities on compressed video packets. Packet selection and scheduling for non-scalably encoded video streams is the focus of this paper.

We propose a generic framework for rate-distortion optimized distributed streaming over a shared communication channel. Though our framework can be applied to any such setting, the paper mainly focuses on the specific example of scheduling multiple video packet streams in a wireless LAN scenario. Using the framework, each of the senders individually allocates a portion of the available bandwidth according to the importance of its packets such that the end-to-end performance in terms of video quality, over all streams is maximized, under given network constraints. The optimization relies on rate-distortion information that is used to characterize a video packet and that comprises the size of the packet in bits and the importance of the packet in terms of reconstruction distortion for the video stream. In essence, the framework enables the senders to trade-off in a coordinated but still distributed fashion rate and distortion not only over their respective video packets, but also across packets that belong to different video streams. In conjunction with the framework, we propose a strategy for dynamic bandwidth adaptation at each sender.

There is a substantial body of prior work on video streaming over wireless LANs, and over wireless networks in general [3]. However, to the best of our knowledge, rate-distortion optimized distributed streaming of multiple video sources as studied in the present paper has not been investigated before. The most closely related contemporaneous works are the following. [4] proposes a cross-layer ARQ algorithm for video streaming in 802.11 wireless networks which gives priority to perceptually more important packets at (re)transmission. Only a single video stream is considered. In [5], a transmission strategy is examined that provides adaptive quality-of-service (QoS) to layered video for streaming over 802.11 WLANs. Again, only a single video stream is considered and no rate-distortion optimization is performed. Similarly, in [6, 7] hybrid transmission techniques that combine Automatic Repeat reQuest (ARQ) and Forward Error Correction (FEC) are proposed for improved real-time video transport over WLANs.

II. PRELIMINARIES

A. Rate-Distortion Characterization

Let k be the index of a packet from a video stream. Then, the rate-distortion (R-D) information associated with packet k consists of the size of packet k in bits $R(k)$ and the importance of packet k for the reconstruction distortion of the video stream denoted as $D(k)$. Specifically, $D(k)$ is the total increase in MSE distortion that will affect the video stream if packet k is not delivered to the receiver on time ($t_{d,k}$), and is computed as $D(k) = \sum_{i=1}^{L} \Delta d_i^k$, where L is the number of packets in the stream and Δd_i^k is the increase in MSE distortion associated with packet i given that packet k is missing at the receiver. In addition, $t_{d,k}$ is the delivery deadline by which packet k must arrive at the receiver in order to be usefully decoded. Note that $\Delta d_i^k = 0$ for $i < k$. Here, we assume that the decoder employs error concealment to account for missing packets.

B. Packet Loss and Delay Probabilities

We model each direction of the network path between a sender/user on the shared channel and its respective receiver as a time-invariant packet erasure channel with random delays. For the forward (uplink) direction to the receiver via the access point, this means that if a sender transmits a data packet at time t, then the packet is lost with

some probability, say ϵ_F, independently of t. However, if the packet is not lost, then it arrives at the receiver at some later time t', where the forward trip time $FTT = t' - t$ is randomly drawn according to a probability density p_F. The backward (downlink) direction from the receiver via the access point to the user is similarly characterized by the probability of packet loss ϵ_B and delay density p_B. Then, these induce the probability $\epsilon_R = 1 - (1 - \epsilon_F)(1 - \epsilon_B)$ of losing a packet in either the forward or backward direction, and the round trip time distribution $P\{RTT > \tau\} = \epsilon_R + (1 - \epsilon_R) \int_\tau^\infty p_R(t)dt$, where $p_R = p_F * p_B$ is the convolution of p_F and p_B. Note that $P\{RTT > \tau\}$ is the probability that the user does not receive an acknowledgement packet by time $t + \tau$ for a data packet sent to the receiver at time t.

C. Multiple Access

We assume that a Time Division Multiple Access (TDMA) [8] scheme is employed in order to allow for the multiple users to share the wireless channel. In TDMA, each of the users is dynamically assigned a time slot based on the user's need for throughput. It is only during this time slot that the user can transmit its data. The time slot assignment is done by the access point that supports the WLAN environment. We assume that each of the users reports its true need for throughput as computed by the optimization algorithm.

III. OPTIMIZATION FRAMEWORK FOR DISTRIBUTED STREAMING

Consider that there are N users sending video packets over the shared medium simultaneously. We are interested in finding the best transmission schedules for the video packets of each stream for a given available bandwidth on the shared channel. The problem can be formalized as follows. Assume that user i, for $i = 1, \ldots, N$, has at time instant t a window $\mathcal{W}_i$ of packets that are considered for (re)transmission. Note that $\mathcal{W}_i$ may include in particular packets from earlier transmissions that have not been acknowledged yet by the corresponding receiver and whose delivery deadlines occur after t. The user needs to decide then on omitting/dropping a subset of packets $\boldsymbol{k}^{(i)} = \{k_1, k_2, \ldots, k_P\}$ (if any) from $\mathcal{W}_i$ prior to transmission such that its assigned transmission bandwidth is not exceeded. For example, if the allocated bandwidth is sufficient to transmit all packets from $\mathcal{W}_i$, then $\boldsymbol{k}^{(i)}$ will be an empty set.

Now, the total increase in expected MSE distortion that will affect stream i if $\boldsymbol{k}^{(i)}$ is dropped prior to transmission can be computed as:

$$\widetilde{D}(\boldsymbol{k}^{(i)}) = \sum_{j \in \boldsymbol{k}^{(i)}} E[D(j)] + \sum_{j \in \mathcal{W}_i \setminus \boldsymbol{k}^{(i)}} E[D(j)], \qquad (1)$$

where "\" denotes the operator "set difference". Note that the above model assumes additivity of the distortions associated with the individual dropped packets, ignoring any interdependencies between their effects on the distortion, which does not necessarily hold true when the dropped packets are not spaced sufficiently far apart with respect to the intra-refresh period. Furthermore, note that in (1) we had to deal with expectations rather than with the actual distortion values because of the random channel effects. In particular, a packet sent over the channel may not necessarily arrive at its destination on time because of random packet loss or delay experienced during transmission. Therefore, the distortion contribution associated with that packet may not necessarily be zero (despite its transmission) and hence can only be accounted for as an expected value.

The expected distortion term associated with a video packet j in (1) can be computed as

$$E[D(j)] = \begin{cases} D(j)P_0(j) & : & j \in \boldsymbol{k}^{(i)} \\ D(j)P_0(j)P_1(j) & : & j \in \mathcal{W}_i \setminus \boldsymbol{k}^{(i)}, \end{cases} \qquad (2)$$

where P_0 is the probability that a packet does not arrive at the receiver by its delivery deadline due to previous transmissions, if any, and P_1 is the probability that a packet does not arrive at the receiver due to the present transmission. Using the channel models from Section II-B these probabilities can be computed as follows. Let $\{t_1, \ldots, t_M\}$ be the set of previous transmission instances of packet j and let t_p denote the present time. Then, we write

$$\begin{aligned} P_0(j) &= \prod_{m=1}^{M} P\{FTT > t_{d,j} - t_m | RTT > t_p - t_m\}, \\ P_1(j) &= P\{FTT > t_{d,j} - t_p\}. \end{aligned} \qquad (3)$$

Finally, $R(\mathcal{W}_i \setminus \boldsymbol{k}^{(i)}) = \sum_{j \in \mathcal{W}_i \setminus \boldsymbol{k}^{(i)}} R(j)$ represents the corresponding average transmission rate of user i over the window $\mathcal{W}_i$.

We denote the available bandwidth of the shared channel as R^*. The total transmission rate of all users should not exceed this quantity, i.e., $R(\boldsymbol{k}) = \sum_{i=1}^{N} R(\mathcal{W}_i \setminus \boldsymbol{k}^{(i)}) \leq R^*$. We are interested in minimizing the overall distortion over all streams, given as $\widetilde{D}(\boldsymbol{k}) = \sum_{i=1}^{N} \gamma(i)\widetilde{D}(\boldsymbol{k}^{(i)})$, such that the constraint on the total transmission rate is satisfied, where $\gamma(i)$ is the weighting factor for stream i that depends on the user's policy. (For example, $\gamma(i) > 1$ may signify that stream i is more important and that therefore should be given a priority.) In other words, we would like to solve for the optimal vector of dropping patterns

$$\boldsymbol{k}^* = \operatorname*{arg\,min}_{\boldsymbol{k}\,:\,R(\boldsymbol{k}) \leq R^*} \widetilde{D}(\boldsymbol{k}), \qquad (4)$$

where $\boldsymbol{k} = (\boldsymbol{k}^{(1)}, \ldots, \boldsymbol{k}^{(N)})$. We solve for the individual optimal drop patterns $\boldsymbol{k}^{(i)*}$ by casting (4) as a non-constrained optimization problem using a Lagrange multiplier ($\lambda > 0$):

$$\boldsymbol{k}^{(i)*} = \operatorname*{arg\,min}_{\boldsymbol{k}^{(i)} \in \mathcal{W}_i} \widetilde{D}(\boldsymbol{k}) + \lambda R(\boldsymbol{k}), \quad i = 1, \ldots, N. \qquad (5)$$

It can be shown that the solution to (5) reduces to dropping every packet $j \in \mathcal{W}_i$ for a sender i such that $\lambda_j \leq \lambda$, where $\lambda_j = \gamma(i)P_0(j)D(j)/R(j)$ is defined as the distortion per unit rate utility for packet j. The rest of the packets from $\mathcal{W}_i$ are transmitted. Hence, we have a distributed strategy where each user decides on which of his own packets should be transmitted such that the end-to-end distortion over all streams is minimized, while at the same time the constraint on the overall transmission rate is satisfied.

The appropriate value of the Lagrange multiplier λ that corresponds to R^* and that should be common among the senders can be computed by each one of them independently using methods such as the bisection search or gradient descent. However, these techniques are iterative and would require recursive running of the optimization algorithm until an appropriate value for λ is found. This in turn would incur excess computation on the side of each sender. Therefore, as an alternative we propose for this scenario to track the value of λ over time as follows. Let t_k, for $k = 0, 1, \ldots$, be the current transmission instance at which the users have just ran the optimization algorithm and let $R_i(t_k)$ be the corresponding transmission rate computed by user i. Then, the value of λ that is used in (5) at the next transmission opportunity (t_{k+1}) is computed as

$$\lambda_{k+1} = \left(\lambda_k + \theta \left(\sum_{i=1}^{N} R_i(t_k) - R^* \right) \right)^+, \qquad (6)$$

where θ is a small constant and the function $(x)^+$ is equal to x, for $x > 0$, and to zero, otherwise. Note that (6) increases the value of λ if the current transmission rate of all users is above R^*, and vice-versa. When λ is increased, the number of packets that are omitted at each

sender is also appropriately increased, thereby causing a reduction in the transmission rate. When λ is decreased, the opposite effect is achieved. Hence, in this way starting from an initial conservative choice for λ each user is provided with a simple control strategy to accordingly adjust its value over time.

IV. SIMULATION RESULTS

In this section, we examine via simulation experiments the performance of the proposed framework for rate-distortion optimized distributed streaming denoted henceforth *RDOpt*. We measure performance in terms of the average luminance (Y) PSNR in dB of the decoded video frames both individually at each receiver and also jointly over all receivers as a function of different channel parameters, namely, available data rate and packet loss rate. In particular, two cases are studied in this context. In the first one, there is sufficient data rate available on the shared channel to transmit every packet of each video stream once, however the network is lossy and some of the transmitted packets are lost. Hence, the senders needs to decide at each transmission opportunity whether (1) to retransmit a previous lost packet, or (2) to transmit a new packet which has not been transmitted before. In the second case, we examine streaming performance when simultaneously the available data rate on the channel can be variable and the channel exhibits random packet loss and delay. Finally, at the end we examine the performance of the algorithm for tracking the Lagrange multiplier at each user proposed in Section III.

The video sequences used in the experiments are coded using JM 2.1 of the JVT/H.264 video compression standard [9]. Four standard test sequences in QCIF format are used: Foreman, Carphone, Mother & Daughter, and Salesman. In other words, the number of users/streams sharing the wireless channel is $N = 4$. Each sequence is encoded at a frame rate of 30 fps and an average Y-PSNR of about 36 dB. The specific rate-distortion encoding characteristics for the four sequences are shown in Table I. The first frame of each sequence is intra-coded, followed by all P-frames. Every 4 frames a slice is intra updated to improve error-resilience by reducing error propagation (as recommended in JM 2.1), corresponding to an intra-frame update period of $M = 4 \times 9 = 36$ frames. An identical importance weight $\gamma = 1$ is applied across all streams.

Sequence	Rate (Kbps)	Y-PSNR (dB)
Foreman	157.45	35.69
Carphone	171.30	36.60
Mother & Daughter	63.79	36.21
Salesman	64.31	35.01

TABLE I
ENCODING CHARACTERISTICS OF THE FOUR SEQUENCES.

We also study the performance of a conventional system for distributed streaming denoted as *Baseline*, which does not consider the distortion importance of different packets. In particular, when making transmission decisions, *Baseline* does not distinguish between two packets related to two different P-frames, except for the size of the packets. Therefore, *Baseline* randomly chooses between two P-frame packets of the same size, for example, when it needs to reduce the number of transmitted packets in order to adapt to the allocated portion of the available bandwidth. Similarly, transmissions of new packets and retransmissions of old lost packets are also performed in a random order by this system. In both systems, *RDOpt* and *Baseline*, each user considers packets for transmission in non-overlapping windows of size 25.

A. Adapting to Packet Loss

In this scenario, the uplink (forward) channel to the access point exhibits random packet loss caused by dropping corrupted packets at the access point, which in turn is due to the presence of a non-zero bit error rate on the uplink channel. Therefore, the users need to decide whether they would retransmit previous lost packets or instead transmit new packets which have not been transmitted yet. In other words, in addition to the packets from the current transmission windows, the senders also consider for the present transmission past packets from previous transmission windows that have been lost during transmission. These experiments assume an ideal feedback channel, i.e., a sender is immediately notified of each lost packet, that the forward channel exhibits no packet delay, and that successive packet losses are independent and identically distributed.
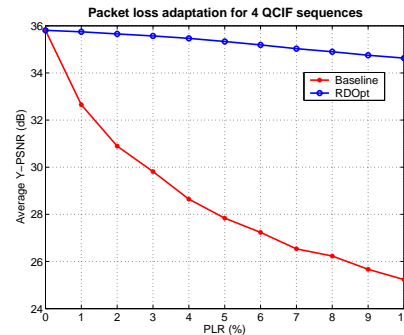

Fig. 1. Y-PSNR (dB) vs. Packet loss rate (%).

Figure 1 shows the overall performances of *RDOpt* and *Baseline* over all four sequences as a function of the packet loss rate (PLR) measured in percent. It can be seen that *RDOpt* provides substantial gains over *Baseline* over the whole range of values considered for the PLR (except of course for PLR = 0%). For example, at packet loss rate of 5%, the performance improvement due to the optimized transmission decisions is 5.5 dB, which is quite impressive. The improved performance is due to the fact that *RDOpt* exploits the knowledge about the effect of loss of individual video packets on the reconstructed video quality. Therefore, under *RDOpt* the users preferentially (re)transmit packets from their transmission windows that are most important for the reconstruction quality of the corresponding video streams.

B. Adapting to Packet Loss and Available Bandwidth

This section investigates the end-to-end performance for the scenario where the available data rate can be varied and the channel exhibits random packet loss and delay on both forward and backward directions. Now, each sender considers packets for transmission in a sliding window of size 10 packets. For every arriving packet on the forward channel the receiver returns immediately to the sender an acknowledgement packet on the backward channel. At each transmission opportunity *Baseline* considers for retransmission only those packets from the transmission window whose last transmission has not been acknowledged within $\mu_R + 3\sigma_R$ seconds from the current transmission opportunity, where μ_R and σ_R are respectively, the mean and the standard deviation of the round-trip time. The play-out delay for each of the videos is 500 ms, and the time interval between transmission opportunities is 33 ms.

The forward and backward channels are modeled as follows. Packets transmitted on these channels are dropped at random, with a drop rate $\epsilon_F = \epsilon_B = \epsilon = 3$ %. Those packets that are not dropped

experience a random delay, where the forward and backward delay densities p_F and p_B are modeled as shifted Gamma distributions with parameters (n, α) and right shift κ. These parameters are estimated from actual traces of packet losses and packet delays collected in wireless LANs.
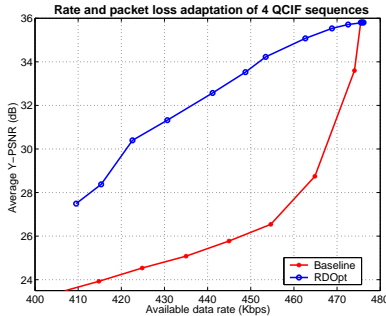
Fig. 2.   Y-PSNR (dB) vs. Available data rate (Kbps).

Figure 2 shows the overall Y-PSNR (dB) performances of *RDOpt* and *Baseline* over all four sequences as a function of the available data rate (Kbps) on the shared channel. It can be seen that also in this case *RDOpt* outperforms *Baseline* with quite a significant margin over the whole range of values considered for the available data rate. For example, at data rates of 450 Kbps, the performance improvement due to the optimized scheduling decisions is around 8 dB. The improved performance of *RDOpt* is due to exploiting the knowledge of the importance of the video packets, as discussed earlier.

Unfortunately, due to lack of space we are not able to include here the performances for the individuals sequences (receivers) in each of the two scenarios that we considered. In essence, these results show that *RDOpt* performs (re)transmission prioritization not only among packets of a video stream, but also across packets of different streams. This is due to the fact that each stream provides a different rate-distortion trade-off, which *RDOpt* takes advantage of.

C. Tracking the right λ and rate control

In this section, we examine the performance of the technique proposed in (6) to track the value of the Lagrange multiplier λ at a sender. We examine the case when a user is added to the system. In particular, the data rate constraint is 380 Kbps, and we have three users active in the system sending respectively, Foreman, Carphone, and Mother & Daughter. Then, at time $t = 50$ seconds a fourth user joins the network and starts transmitting the fourth video used in our experiments, Salesman. We examine how the system allocates rates to the users after the new user joins in. Note that prior to the increase in number of users the overall date rate available on the channel is approximately sufficient to send all three streams at their encoding rates. This can be easily verified from Table I. However, after the fourth users starts sending video packets, the system needs to adjust to the new situation and to reallocate data rates to each user accordingly.

In Figure 3 we show the variations of allocated data rates over time. It can be seen that after the fourth user joins the network, it starts to increase gradually its date rate on the shared channel. However, the system is quick to learn that in the new situation there is an insufficient data rate to allow everyone to transmit at their encoding rates. Therefore, the Lagrange multiplier is accordingly increased and varied until a new equilibrium point is reached over time. Note that the reallocated data rates actually affect only the last two users as
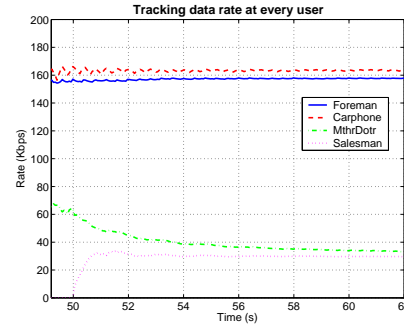
Fig. 3.   Tracking date rates when a fourth user joins in at $t = 50$ sec.

shown in Figure 3. In particular, the system simply re-allocates to the new user (Salesman) some of the data rate assigned previously to the user Mother & Daughter. This behaviour was seen throughout the experimental results reported in this paper and in essence it is due to the different importance of the video packets for the reconstruction quality of each stream, as explained earlier.

V. CONCLUSIONS

A framework for rate-distortion optimized distributed streaming of multiple video sources over a shared communication channel has been presented. The proposed framework enables the users to perform optimal transmission decisions so that the overall video quality across all streams is maximized for the given available data rate on the shared channel. The framework relies on rate-distortion hint track information that describes a video packet. We have examined the performance of our framework for two canonical problems in video streaming: packet loss adaptation and simultaneous bandwidth and packet loss adaptation. Significant gains in performance ranging up to 8 dB are registered in both of these scenarios over a conventional system for distributed streaming which does not take into account the distortion information associated with the video packets. Finally, in conjunction with the framework we have proposed and examined the performance of a simple tracking scheme for adaptively controlling the data rate at which individual users can transmit on the channel.

REFERENCES

[1] "Nielsen//NetRatings, Inc." http://www.nielsen-netratings.com/.
[2] B.-J. Kim, Z. Xiong, , and W. A. Pearlman, "Low bit-rate scalable video coding with 3D set partitioning in hierarchical trees (3-D SPIHT)," *IEEE Trans. CSVT*, vol. 10, no. 8, pp. 1374–1387, Dec. 2000.
[3] M. Etoh and T. Yoshimura, "Advances in wireless video delivery," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 111–122, Jan. 2005.
[4] P. Bucciol, G. Davini, E. Masala, E. Filippi, and J. D. Martin, "Cross-layer perceptual ARQ for H.264 video streaming over 802.11 wireless networks," in *Proc. Global Tellecomunications Conference (Globecom)*, vol. 5. Dallas, TX, USA: IEEE, Nov./Dec. 2004, pp. 3027–3031.
[5] Q. Li and M. van der Schaar, "Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 278–290, Apr. 2004.
[6] A. Majumdar, D. G. Sachs, I. V. Kozintsev, K. Ramchandran, and M. M. Yeung, "Multicast and unicast real-time video streaming over wireless LANs," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 6, June 2002.
[7] M. Chen and G. Wei, "Multi-stages hybrid ARQ with conditional frame skipping and reference frame selecting scheme for real-time video transport over wireless LAN," *IEEE Trans. on Consumer Electronics*, vol. 50, no. 1, pp. 158–167, Feb. 2004.
[8] K. Pahlavan and P. Krishnamurthy, *Principles of Wireless Networks: A Unified Approach*, Englewood Cliffs, NJ: Prentice-Hall, Dec. 2001.
[9] Telecom. Standardization Sector of ITU, "Video coding for low bitrate communication," *Draft ITU-T Recommendation H.264*, Mar. 2003.