Rate-Distortion Optimized Distributed Packet Scheduling of Multiple Video Streams Over Shared Communication Resources

Jacob Chakareski and Pascal Frossard, *Senior Member, IEEE*

***Abstract*—We consider the problem of distributed packet selection and scheduling for multiple video streams sharing a communication channel. An optimization framework is proposed, which enables the multiple senders to coordinate their packet transmission schedules, such that the average quality over all video clients is maximized. The framework relies on rate-distortion information that is used to characterize a video packet. This information consists of two quantities: the size of the packet in bits, and its importance for the reconstruction quality of the corresponding stream. A distributed streaming strategy then allows for trading off rate and distortion, not only within a single video stream, but also across different streams. Each of the senders allocates to its own video packets a share of the available bandwidth on the channel in proportion to their importance. We evaluate the performance of the distributed packet scheduling algorithm for two canonical problems in streaming media, namely adaptation to available bandwidth and adaptation to packet loss through prioritized packet retransmissions. Simulation results demonstrate that, for the difficult case of scheduling nonscalably encoded video streams, our framework is very efficient in terms of video quality, both over all streams jointly and also over the individual videos. Compared to a conventional streaming system that does not consider the relative importance of the video packets, the gains in performance range up to 6 dB for the scenario of bandwidth adaptation, and even up to 10 dB for the scenario of random packet loss adaptation.**

***Index Terms*—Distributed systems, multimedia streaming, optimization, packet scheduling, rate-distortion, video communication, wireless networks.**

I. INTRODUCTION

THE DEMAND for multimedia traffic sent over the Internet exhibits an ever growing trend today [1], [2]. 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, or through a common bottleneck network node in the Internet, are typical instances of such scenarios. In that context, it becomes important to consider the performance of the whole streaming system, in order to maximize the overall quality of service of all users. The multiple media sources therefore have to be considered jointly, and only a concerted streaming policy can lead to minimal average distortion. The streaming strategy is either activated at the bottleneck network node, or even better in a distributed manner among the video sources that finely adapt their packet streams to the available communication resources.

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 the timing constraints of the underlying streaming application. Now, randomly omitting packets can have an unpredictable effect on the reconstruction quality of a video stream at the receiver.

Solutions may be proposed that try to adapt the representation of the video information to streaming resource variations, at the price however of high complexity, or loss in coding performance. Video transcoding [3]–[6], for example, re-encodes the stream in order to adapt the bit rate to the available resource, but it is quite greedy in terms of complexity. Scalable coding techniques [7]–[11] have been developed to solve these problems, where the scalable encoding provides an inherent prioritization among the compressed data. This offers 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, nonscalable or nonprioritized 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. Adaptive streaming via efficient selection and scheduling of nonscalably encoded video packets is the focus of the present paper.

We propose a generic framework for rate-distortion (R-D) optimized distributed streaming over a shared communication channel. While our framework can be applied to any such settings, the paper mainly focuses on the specific example of scheduling multiple video packet streams in a wireless LAN scenario. Each of the senders individually allocates a portion of the available bandwidth to its respective video stream such that the end-to-end performance in terms of video quality, over all streams, is maximized, under the given network constraints. The framework relies on R-D information that is used to characterize a video packet. It basically consists of two quantities: the size of the packet in bits, that is usually available in packet headers, and the importance of the packet in terms of the reconstruction distortion for the video stream. In essence, the framework enables the senders to trade-off in a coordinated but still distributed

Manuscript received March 7, 2005; revised September 26, 2005. This work was supported by the Swiss National Science Foundation, under Grant PP-002-68737. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Rynson W. H. Lau.

The authors are with the Signal Processing Institute, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland (e-mail: jakov.cakareski@epfl.ch; pascal.frossard@epfl.ch).

Digital Object Identifier 10.1109/TMM.2005.864284

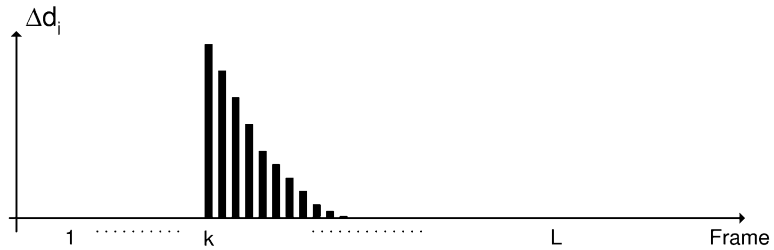

Fig. 1. Loss of frame k induces distortion in later frames. $D(k)$ is the total distortion summed over all affected frames.

fashion rate and distortion not only over their respective video packets, but also across packets that belong to different video streams. The main contributions of the present paper are to extend the optimization framework from [12] to the scenario of distributed streaming of multiple video streams, and to examine the specific challenges that arise therein.

There is a substantial body of prior work on video streaming over wireless LANs, and over wireless networks in general [13]. However, to the best of our knowledge, R-D 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. [14] 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 [15], 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 R-D optimization is performed. Similarly, in [16], [17] hybrid transmission techniques that combine Automatic Repeat reQuest (ARQ) and Forward Error Correction (FEC) are proposed for improved real-time video transport over WLANs. In addition, in [18], the authors propose a system that combines R-D optimized data partitioning and prioritized adaptive (re)transmission for robust streaming of a single video source over a wireless LAN. Similarly, the authors in [19] introduce a cross-layer protection strategy that combines adaptive application-layer FEC and physical-layer modulation with fine-granular-scalability (FGS) coding to improve the resilience of wireless video transmission. Finally, our work is perhaps most closely related to [20], [21], which study R-D optimized bandwidth adaptation of multiple incoming video streams at a network node. Here, the node performs centralized optimal packet dropping across the multiple streams in order to adjust the data rate on the outgoing link. However, the proposed streaming techniques do not extend to the case of distributed scheduling of concurrent media streams, which is the scenario studied in the present paper.

The rest of the paper is structured as follows. In the next section, we present the R-D information that is associated with a video packet and our abstraction of the network path between a user on the shared channel and its respective receiver (that can be located potentially anywhere in the Internet). These characterizations of the video source and the communication channels are employed by the optimization framework, introduced in Section III to perform transmission decisions for the packets of every video stream that are optimal in an R-D sense. Then, in Section IV, we discuss two possible applications of our optimization framework, namely, bandwidth adaptation of multiple incoming video streams at a network node, and streaming multiple video sources over a wireless LAN. Next, in Section V we examine the performance of our framework and we compare it to that of a conventional system for distributed video streaming over a shared communication channel. Finally, concluding remarks are provided in Section VI.

II. Preliminaries

A. Rate-Distortion Characterization

Let k be the index of a packet from a video stream. Then, the 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 $D(k) = \sum_{i=1}^{L} \Delta d_i$, where L is the number of packets in the stream and Δd_i 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 $\Delta d_i = 0$ for $i < k$. In Fig. 1 we illustrate the distortions Δd_i for the loss of packet k, where for clarity of presentation it is assumed that each packet i corresponds to a video frame.

It can be seen that the MSE per frame ramps up at frame k, which is expected since the missing frame k is replaced with frame $k-1$ and there are no prior losses. Here, we assume that previous frame concealment is used for missing frames. Due to error propagation, which in turn is caused by the predictive nature of the encoding process, the MSE associated with subsequent frames also exhibits a nonzero value, as shown in Fig. 1. However, due to the effects of spatial filtering and intra refresh [22], its amplitude gradually decreases over successive frames, till it finally becomes zero at frame $j > k$ sufficiently apart from k.

Note that in live streaming scenarios, where video content is created on the fly, a sender would not have access at any instance to all the packets from the video stream that it is transmitting[1]. That is simply because some of the video packets will be created (encoded) in the future, i.e., after a particular transmission instance. Therefore, the number of packets L that a sender can use in this context to compute the distortion information $D(k)$

[1]In other words, the size of the pre-fetch window of media packets available to the sender is quite small.

associated with a packet k would actually refer to the number of successive packets available in the sender's buffer at transmission time of packet k.

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 $\mathrm{FTT} = t' - t$ 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 $P\{\mathrm{RTT} > \tau\} = \epsilon_R + (1-\epsilon_R)\int_\tau^\infty p_R(t)dt$, where $p_R = p_F * p_B$ is the convolution of p_F and p_B. Note that $P\{\mathrm{RTT} > \tau\}$ is the probability that the user does not receive an acknowledgment packet by time $t+\tau$ for a data packet sent to the receiver at time t. The properties of the shared channel in terms of packet loss and packet delay are included in the model described above, as the shared channel represents a segment of the network path between a sender and a receiver.

III. Optimization Framework for Distributed Streaming

A. Expected Distortion and Rate

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.f

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

$$\begin{aligned}\tilde{D}(\boldsymbol{k}^{(i)}) &= \sum_{j\in\mathcal{W}_i} E[D(j)] \\ &= \sum_{j\in\boldsymbol{k}^{(i)}} D(j)P_0(j) + \sum_{j\in\mathcal{W}_i\setminus\boldsymbol{k}^{(i)}} D(j)P_0(j)P_1(j),\end{aligned} \tag{1}$$

where "$\backslash$" denotes the operator "set difference". 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

$$\begin{aligned}P_0(j) &= \prod_{m=1}^{M} P\{\mathrm{FTT} > t_{d,j} - t_m | \mathrm{RTT} > t_p - t_m\}, \\ P_1(j) &= P\{\mathrm{FTT} > t_{d,j} - t_p\}.\end{aligned} \tag{2}$$

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, as recognized for example in [23]. Still, due to its simplicity and yet good accuracy, the additive model has found a number of applications in streaming and modeling of packetized media, such as [12], [24], [25]. 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.

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 $\mathcal{W}_i$.

B. Problem Formulation

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., $R(\boldsymbol{k}) = \sum_{i=1}^N R(\mathcal{W}_i \setminus \boldsymbol{k}^{(i)}) \le R^*$. We are interested in minimizing the overall distortion over all streams, given as $\tilde{D}(\boldsymbol{k}) = \sum_{i=1}^N \gamma(i)\tilde{D}(\boldsymbol{k}^{(i)})$, 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[2]. In other words, we would like to solve for the optimal vector of dropping patterns

$$\boldsymbol{k}^* = \arg\min_{\boldsymbol{k}:R(\boldsymbol{k})\le R^*} \tilde{D}(\boldsymbol{k}) \tag{3}$$

where $\boldsymbol{k} = (\boldsymbol{k}^{(1)},\ldots,\boldsymbol{k}^{(N)})$. We solve for the individual optimal drop patterns $\boldsymbol{k}^{(i)*}$ by casting (3) as a nonconstrained optimization problem using a Lagrange multiplier ($\lambda > 0$):

$$\boldsymbol{k}^{(i)*} = \arg\min_{\boldsymbol{k}^{(i)}\in\mathcal{W}_i} \tilde{D}(\boldsymbol{k}) + \lambda R(\boldsymbol{k}), \quad i = 1,\ldots,N. \tag{4}$$

It can be shown that the solution to (4) reduces to dropping every packet $j \in \mathcal{W}_i$ for a sender i such that $\lambda_j \le \lambda$, where

[2]For example, $\gamma(i) > 1$ may signify that stream i is more important and that therefore should be given a priority.

$\lambda_j = \gamma(i)P_0(j)D(j)/R(j)$ 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.

Finally, it should be mentioned that the objective function $\tilde{D}(\boldsymbol{k})$ of the optimization algorithm as defined above represents only one possible choice. In particular, we decided to define $\tilde{D}(\boldsymbol{k})$ as a weighted sum of the individual distortions over all streams as we were interested in maximizing the overall R-D performance of the scheduling system. In practice, sometimes one may be interested in defining a different objective function, for example the maximum distortion over all streams, in which leads to a min-max optimization problem. Nonetheless, the generality of the optimization framework as presented thus far allows handling multiple choices for the objective function of interest without prior modification of the framework.

C. Computation of the Lagrange Multiplier λ

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 the distributed 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 (4) at the next transmission opportunity (t_{k+1}) is computed as

$$\lambda_{k+1} = \left(\lambda_k + \theta\left(\sum_{i=1}^{N} R_i(t_k) - R^*\right)\right)^+ \qquad (5)$$

where θ is a small constant and the function $(x)^+$ is equal to x, for $x > 0$, and to zero, otherwise. Note that (5) 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.

Finally, it should be mentioned that (5) represents an instance of the sub-gradient method. This class of methods are typically used when Lagrange relaxation is invoked in optimization problems with integer constraints. Their properties have been studied in greater detail, for example in [26]. In addition, in a recent work [27] on Internet pricing for general data services the authors provide analysis that among others argues stability and convergence of adaptation algorithms such as (5).

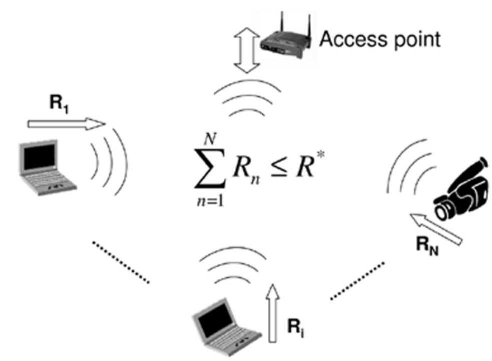

Fig. 2. N video streams sharing a wireless channel.

IV. R-D Optimized Distributed Streaming

In this section, we consider two streaming scenarios where the generic optimization framework proposed in Section III can potentially be employed. The first application is scheduling of multiple concurrent streams over a wireless LAN. The second application is bandwidth adaptation via packet dropping at a network node in the Internet. In both of them, the proposed framework is used by the agent(s) in the system to perform transmission decisions for every packet of the involved video streams.

A. Distributed Streaming Over Wireless LANS

We face an increasing proliferation of wireless LANs [28] at present as they provide a flexible and cost effective solution for many applications in computer networking [29]. Therefore, it is natural to expect that multimedia networking over WLANs will gain a momentum in terms of importance both for practical applications and as a research problem. In the scenario considered here, there are multiple sources of video traffic communicating over a shared wireless medium, as illustrated in Fig. 2. The communication is performed via an access point that supports the WLAN environment. Using the proposed optimization framework, each of the sources can then independently optimize the transmission schedule for its own packets such that the video quality over all streams sent over the shared channel is maximized.

1) Application to TDMA systems: We assume that a time division multiple access (TDMA) 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 and we assume that each of the users reports its true need for throughput as computed by the optimization algorithm. TDMA schemes have been used in several variations of WLANs, such as HiperLAN/2 [30], Bluetooth [31], and home RF networks.

2) Extension to CSMA/CA: We now present a variation of the proposed optimization framework for the case when an alternative scheme known as carrier sense multiple access with collision avoidance (CSMA/CA) is employed for sharing the wireless medium among the multiple users. With CSMA/CA, the users have to contend first for using the communication channel prior to their actual transmissions. CSMA/CA has been predominantly used in the series of IEEE 802.11 WLAN standards [32].

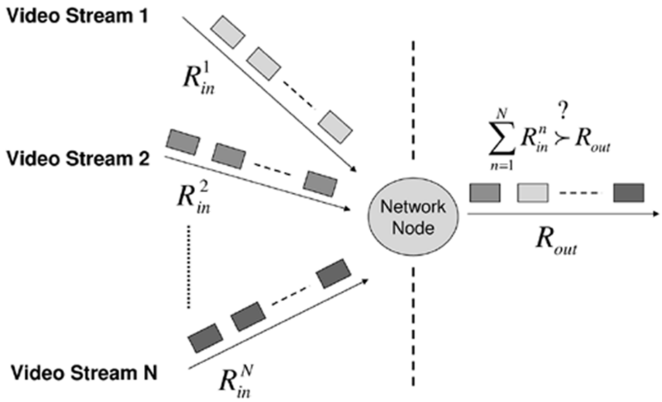

Fig. 3. N incoming video streams at a network node that have to be multiplexed on a single outgoing link.

From their respective pools of packets considered for transmission at present, $(\mathcal{W}_i, i = 1, \ldots, N)$, every user i selects its most important packet j based on the distortion per bit utilities λ_j for $j \in \mathcal{W}_i$. In essence, each of the users finds in its transmission window the packet with the highest utility. Then, the users broadcast these utility values in order to agree on transmission priorities. The utilities are sorted in decreasing order by each user, and each of the users transmits its own most important packet based on this order. After all the packets from the sorted list are transmitted, the users update their transmission windows $\mathcal{W}_i$ and repeat the same procedure. In this way, we ensure that there is still some level of fairness provided to all users. In other words, the proposed transmission protocol will prevent a situation where a single user, who may indeed have many packets with high distortion per unit rate utilities, transmits all of the time exclusively thereby blocking the other users from sending any of their own packets [33].

B. Bandwidth Adaptation via Packet Dropping

This scenario is commonly encountered in the Internet and it occurs whenever the data rate on the incoming link at a network node exceeds the data rate on the outgoing link. Buffer management during transient periods of network congestion when queues overflow and transcoding at the junction point of two heterogeneous (in terms of available bandwidth) networks are two principal examples of bandwidth adaptation. The incoming traffic at the node consists of multiple video streams that are multiplexed by the node on a single outgoing link. Employing the framework from Section III, the distributed streaming system, as represented by the network node, is interested then in optimizing the overall quality over all streams, for the given resources, as represented by the available bandwidth on the outgoing link. The scenario under consideration is illustrated in Fig. 3.

Note that in this setting, it is the network node that computes the optimal schedules for the packets of the incoming streams. In other words, by employing the framework from the previous section and based on the R-D information associated with every incoming packet the node decides which packets from every stream will be dropped at the node due to insufficient bandwidth on the outgoing link, and which ones will be forwarded. In addition to computing the schedules for every stream, the node also computes what is the appropriate Lagrange multiplier λ that should be used in (4), for the given bandwidth R^* on the outgoing link. As explained earlier, this can be done in an iterative fashion using fast convex search techniques, or alternatively by using the tracking method proposed in Section III-C.

V. Simulation Results

In this section, we examine via simulation experiments the performance of the proposed framework for R-D optimized distributed streaming denoted henceforth RDOpt. In the experiments, we focus exclusively on the second prospective application of our framework discussed in Section IV-A, i.e., transmission of multiple video sources over a common wireless channel. However, it is important to note that some results obtained for this particular setting are directly equivalent to the scenario considered in Section IV-B.

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, three scenarios are considered in this context. In the first one, the channel is lossless, but there is insufficient transmission data rate to send all video packets across the channel. Therefore, the senders need to decide which packets to send and which packets to omit/drop. In the second scenario, 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. Finally, the third scenario under consideration represents a combination of the first two with the addition that transmitted packets here that are not lost experience a random delay over the channel. Specifically, in this scenario we examine streaming performance when simultaneously the transmission data rate can be variable and the channel exhibits random packet loss and delay.

In addition, we also examine how the framework performs rate allocation to the individual users as a function of the available date rate on the shared channel. Finally, at the end we examine the performance of the algorithm for tracking the Lagrange multiplier at each user proposed in Section III-C. In particular, we study how through this algorithm the system controls the data rate placed by the users on the channel both in a steady-state operation and in transient scenarios. Typical transient situations are when a new user joins the system or when there is a sudden change in the available data rate on the shared channel.

The video sequences used in the experiments are coded using JM 2.1 of the JVT/H.264 video compression standard [34]. 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 R-D encoding characteristics for the four sequences are shown in Table I. The first

TABLE I
ENCODING CHARACTERISTICS OF THE FOUR SEQUENCES.

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

frame of each sequence is intra-coded, followed by all P-frames. Every four 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.

In the experiments, 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. *Baseline* randomly chooses between two P-frame packets of the same size, when adapting to the allocated portion of the available bandwidth. In both systems, RDOpt and *Baseline*, each user considers video packets for transmission in nonoverlapping windows of size 25.

A. Adapting to Available Bandwidth

In this particular setting, we examine the performance of RDOpt and *Baseline* for the case when the available data rate is insufficient to support transmission at full rate for each user, so the users have to adapt to the allocated bandwidth. *Baseline* allocates portions of the available transmission bandwidth to each user in proportion to the encoding rates of the corresponding video streams of the users.

Fig. 4 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 RDOpt outperforms *Baseline* with quite a significant margin over the whole range of values considered for the available data rate. This is due to the fact that RDOpt exploits the knowledge about the effect of dropping of individual video packets on the reconstructed video quality. Therefore, under RDOpt users drop packets from their transmission windows that will have the least impact on the overall quality of the reconstructed videos. As can be seen from the figure, the performance gains of RDOpt over *Baseline* increase as the available data rate decreases. For example, at data rate of 410 Kbps, the performance improvement due to the optimized packet dropping decisions is around 6 dB, which is quite impressive.

Finally, the allocation of data rates to the individual sequences as a function of the available data rate on the channel for both, RDOpt and *Baseline*, is shown in Fig. 6. It can be seen from the figure that *Baseline* in essence allocates data rates in proportion to the encoding rate of each sequence and independently of the available data rate on the shared channel. On the other

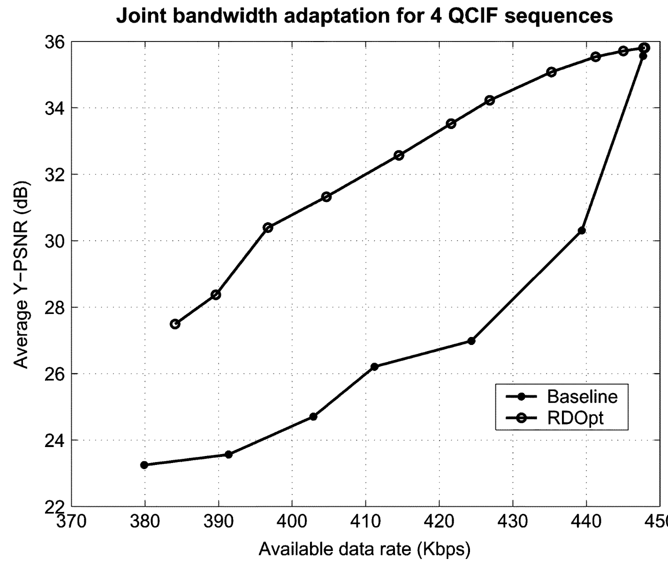

Fig. 4. Y-PSNR (dB) versus data rate (Kbps).

hand, RDOpt assigns increasingly larger shares of the overall rate to Foreman and Carphone, as the data rate is decreased. This is expected and is due to the fact that these two sequences have a more significant impact on the overall performance, as explained earlier. As the data rate is increased, RDOpt gradually decreases the shares allocated to Foreman and Carphone, and increases those for Mother & Daughter, and Salesman, as seen in Fig. 6 (right). This is due to the fact that at these overall data rates there is already enough rate for the former two sequences, so the optimization algorithm can allocate now increasingly more rate to the less important sequences, i.e., the latter two.

Finally, it should be noted that both systems, RDOpt and *Baseline*, will exhibit the same performances as the ones demonstrated here, for the alternative centralized scenario of bandwidth adaptation described in Section IV-B, where a bottleneck network nodeat implements an R-D optimized packet dropping strategy.

Fig. 5 shows the performances of RDOpt and *Baseline* for the individual sequences. It can be seen from the figure that also in this case a significant improvement in performance is observed relative to *Baseline* when packets are dropped in an R-D optimal way. For example, when Mother & Daughter, and Salesman are transmitted over the shared channel at 60 Kbps each, gains of 4 dB are registered over *Baseline*, as shown in the bottom part of Fig. 5. Furthermore, it is interesting to note from the top part of Fig. 5 that no rate reduction and only very little rate reduction are performed by the optimization algorithm for Foreman and Carphone, respectively. In other words, no packets from Foreman and only a few packets from Carphone are dropped. This is because these two sequences exhibit a lot of motion and scene complexity, and therefore will exhibit a significant reduction in quality even for a small number of dropped packets. On the other hand, the sequences Mother & Daughter and Salesman are far less complex in this regards, which means error concealment can be applied quite successfully on their missing packets. Hence, RDOpt trades-off packets from Mother & Daughter, and Salesman for those of Foreman and Carphone in order to maximize the overall performance over all sequences.

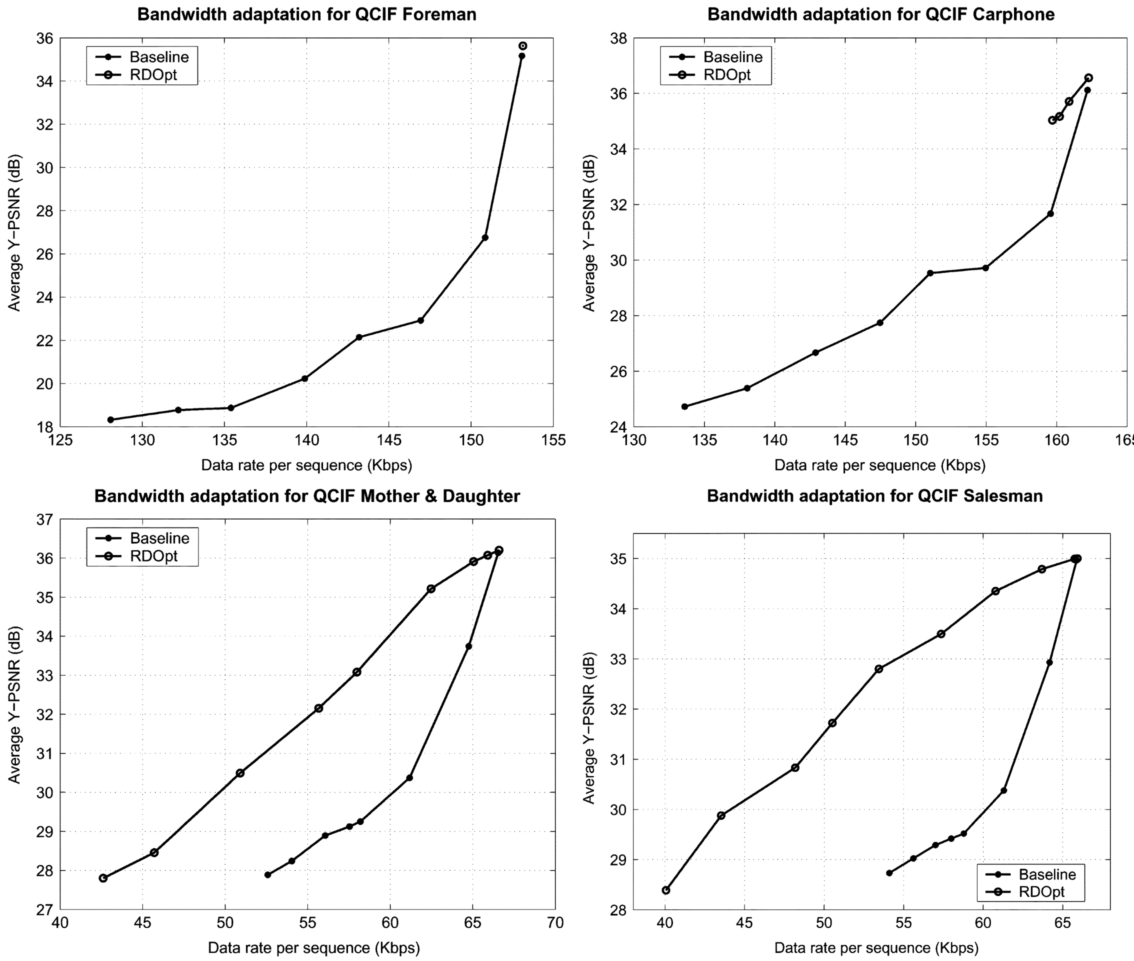

Fig. 5. Y-PSNR (dB) versus data rate per sequence (Kbps) for (top left) Foreman, (top right) Carphone, (bottom left) Mother & Daughter, and (bottom right) Salesman.

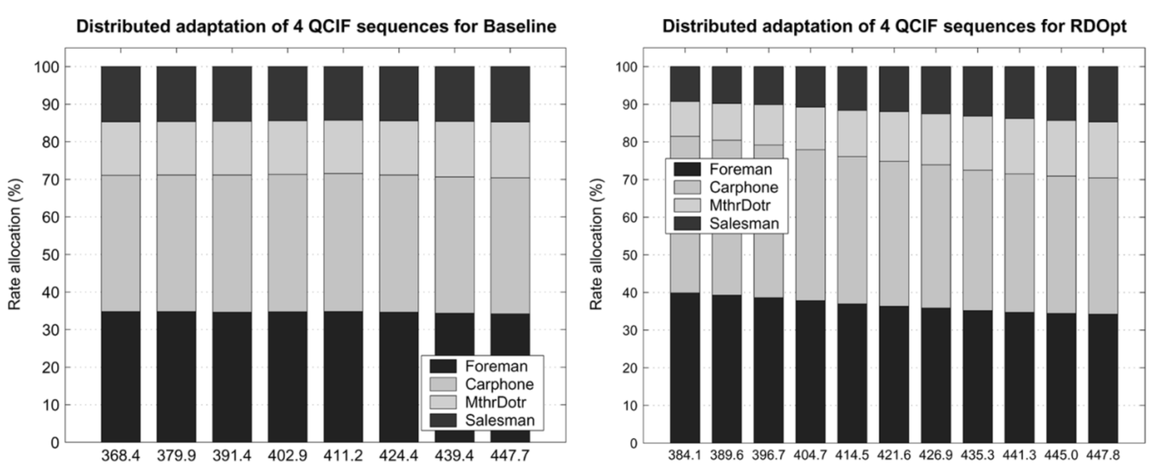

Fig. 6. Allocation (%) of the available data rate (Kpbs) on the outgoing link for (left) *Baseline* and (right) RDOpt.

B. Adapting to Packet Loss

In this scenario, we study the performance of *Baseline* and RDOpt for the case when there is sufficient data rate to allow each user to transmit at the encoding rate of the corresponding video stream. However, now 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 nonzero 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,

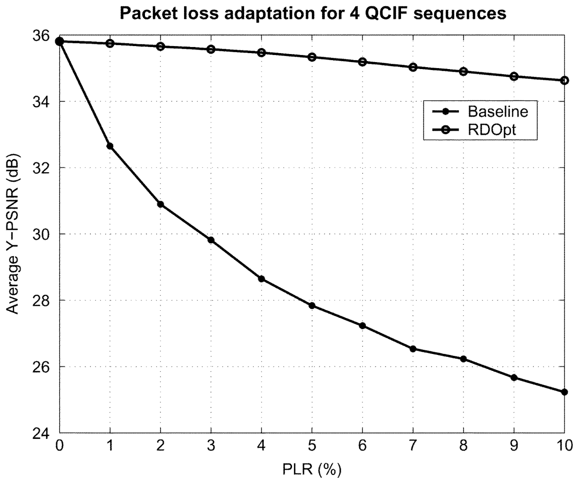

Fig. 7. Y-PSNR (dB) versus packet loss rate (%).

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 that the feedback channel is ideal, 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. 7 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 also in this scenario RDOpt provides substantial gains over *Baseline* over the whole range of values considered for the PLR (except of course for $\mathrm{PLR} = 0\%$). 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, as explained earlier. 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. Note that RDOpt performs (re)transmission prioritization not only among packets of a video stream, but also across packets of different streams, as discussed earlier.

Next, in Fig. 8 we show the performances of RDOpt and *Baseline* for the individual sequences. It can be seen that also across the individual sequences a significant improvement in performance is observed relative to *Baseline* when packet transmission decisions are optimized jointly over the video streams. For example, the gains over *Baseline* at packet loss rate of 5% are 11 dB, 8.5 dB, 2 dB, and 1 dB respectively for Foreman, Carphone, Mother & Daughter, and Salesman. Furthermore, the results from Fig. 8 clearly depict how RDOpt trades-off rate and distortion across the different sequences. Specifically, sufficient data rates are allocated to Foreman and Carphone over the whole range of PLR values under consideration such that all of their packets are delivered to their respective receivers. Note that the allocated data rates include retransmissions of packets lost during prior transmissions. On the other hand, this is not true for Mother & Daughter, and Salesman as evident from their performances shown in the bottom part of Fig. 8. Hence, RDOpt decides to place (re)transmission priority on packets from the former two sequences at the expense of packets from the latter two. The reason for this was explained earlier in the context of bandwidth adaptation in Section V-A.

C. Adapting to Packet Loss and Available Bandwidth

This section investigates the end-to-end performance for the scenario where the available data rate rate can be varied and the channel exhibits random packet loss and delay in both forward and backward directions. 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, courtesy of the authors in [36], [37].

Now, each sender considers packets for transmission in a sliding window of size ten packets. For every arriving packet on the forward channel the receiver returns immediately to the sender an acknowledgment 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 $\mu_R + 3\sigma_R$ s from the current transmission opportunity, where μ_R and σ_R are respectively, the mean and the standard deviation of the round-trip time. This time-out value is frequently used in ARQ systems, e.g., TCP [35]. The play-out delay for each of the videos is 500 ms, and the time interval between transmission opportunities is 33 ms.

Fig. 9 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. The improved performance of RDOpt is due to the same reasons that were discussed earlier. Note that the results presented here are analogous to those from Section V-A except for the fact that both, RDOpt and *Baseline*, have to spend more data rate in this case in order to achieve the same performance relative to the results shown in Fig. 4. This is because now they have to account for the random packet loss that occurs during transmission in each direction. As the streaming results over the individual sequences for this scenario are equivalent to those shown in Fig. 5, except again for the increase in date rate for the same Y-PSNR performance, they are omitted here.

D. Tracking the Right λ and Rate Control

In this section, we examine through several experiments the performance of the technique proposed in (5) to track the value of the Lagrange multiplier λ at a sender. As explained in Section III-C, through the multiplier λ we adaptively control the data rate at each sender as well as the overall data rate of

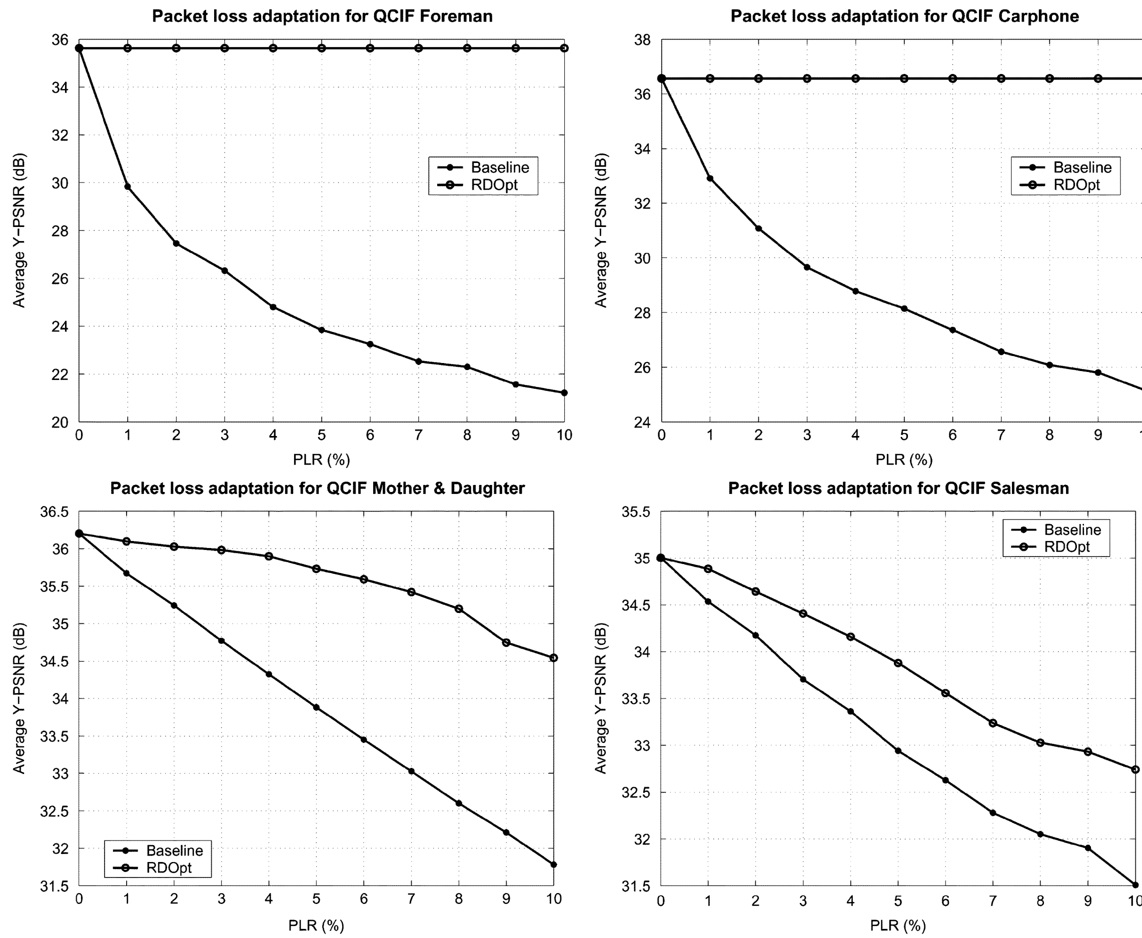

Fig. 8. Y-PSNR (dB) versus Packet loss rate (%) for (top left) Foreman, (top right) Carphone, (bottom left) Mother & Daughter, and (bottom right) Salesman.

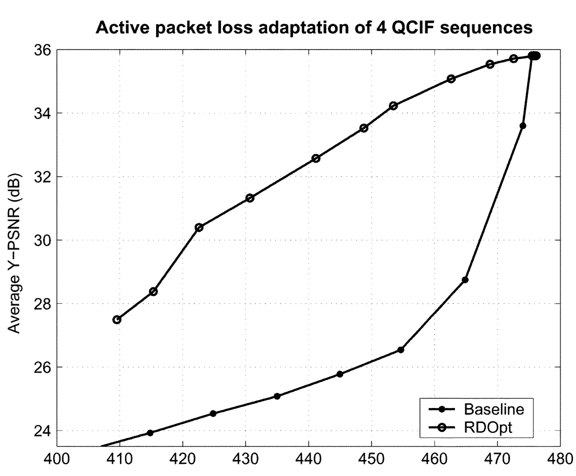

Fig. 9. Y-PSNR (dB) versus data rate (Kbps) for bandwidth and packet loss adaptation.

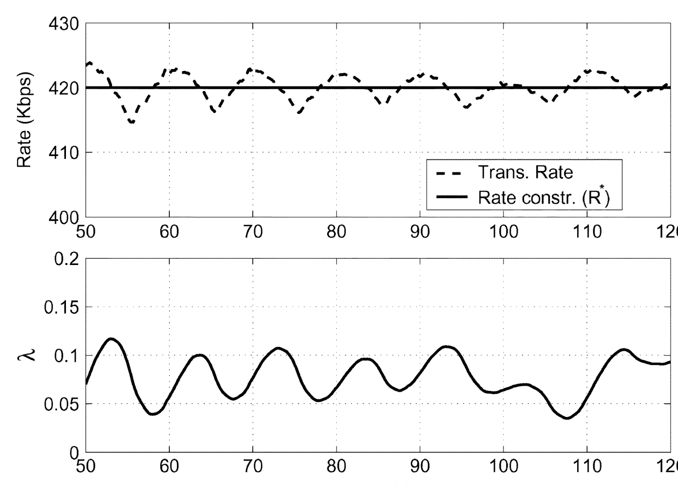

Fig. 10. (Bottom)Tracking λ and (top) the overall data rate over time.

all senders. The value of the multiplier θ that is used in (5) is determined empirically, based on the actual video data that is used in the experiments. In particular, θ is chosen such that it ensures stability and quick convergence of the expression in (5).

First, we consider the performance of (5) for a given data rate constraint R^*. Fig. 10 (bottom) depicts a snapshot of the variations of λ over time for the given rate constraint, while Fig. 10 (top) does the same for the corresponding overall data rate placed by the users on the shared channel. In essence, it can

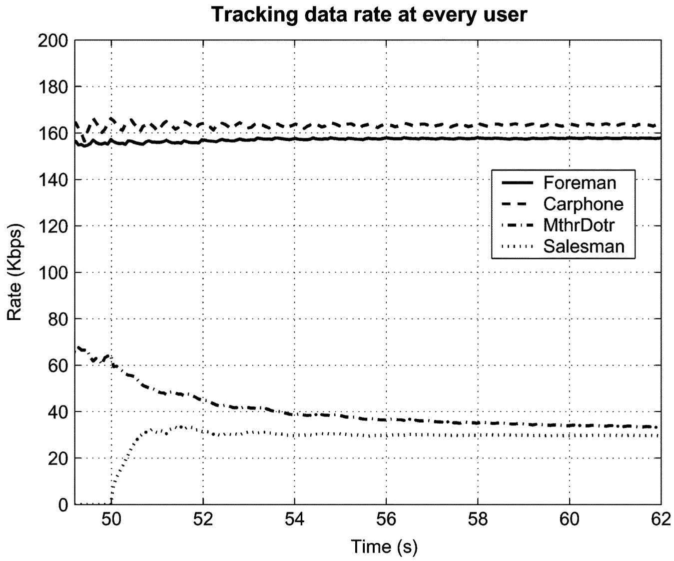

Fig. 11. Tracking date rates when a fourth user joins in at $t = 50$ s.

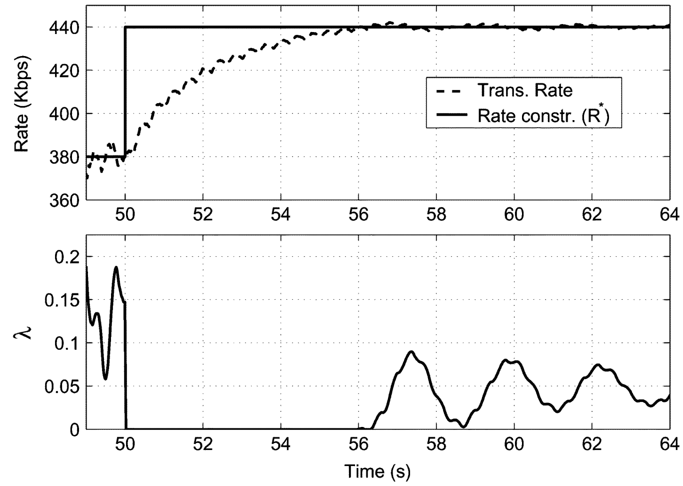

Fig. 12. (Bottom) Tracking λ and the overall date rate (top) when R^* is suddenly increased at $t = 50$ s.

be seen from Fig. 10 that as the overall date rate varies around $R^* = 420$ Kbps due to variations in packet sizes for each video stream, which in turn is caused by the variability in video content over time, the Lagrange multiplier λ is continuously adjusted, i.e., increased or decreased, by (5) in order to control the overall data rate accordingly.

In the next experiment, we examine the performance of the proposed framework 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 $t = 50$ s 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 Fig. 11, we examine 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 shown in Fig. 11. In particular, the system simply re-allocates to the new user (Salesman) some of the data rate assigned previously to the user with the lowest complexity sequence (Mother & Daughter). This behavior was seen throughout the experimental results reported in this paper and in essence is due to the different importance of the video packets for the reconstruction quality of each stream, as explained earlier.

Finally, in Fig. 12 we examine the performance of the system when there is a variation in the available overall data rate. In particular, at time $t = 50$ s the available data rate R^* is increased from 380 to 440 Kbps. As shown in Fig. 12 (top) the system learns about the increase in overall bandwidth and adjusts over the course of a few seconds the date rates of each user such that total data rate placed on the network by all users stays in the vicinity of the new rate constraint. This is achieved by accordingly adapting the Lagrange multiplier λ using (5) as seen from Fig. 12 (bottom). In particular, at the advent of increase in bandwidth λ simply decreases to zero. This allows for the users to increase their data rates as now there is more room on the network for their packets. Then, as their overall data rate reaches the new rate constraint R^*, λ again becomes nonzero and varies in order to ensure that the users transmit at an overall date rate that stays in the vicinity of the new R^*, as shown in Fig. 12.

VI. Conclusions

A framework for R-D optimized distributed streaming of multiple video sources over a shared communication channel has been presented. The framework has been particularly investigated for the case when a TDMA scheme is employed to allow simultaneous channel access to multiple users, and a possible extension of the framework has been discussed for the case when an alternative CSMA/CA scheme is used for the same purpose. 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 employs an R-D hint track information that describes a video packet in order to perform optimal transmission decisions. The hint track information comprises the size of the packet in bits, and the importance of the packet for the reconstruction quality of the corresponding stream. We have examined the performance of our framework for two canonical problems in video streaming: bandwidth adaptation and packet loss adaptation. Significant gains in performance on the order of several dBs, both jointly over all the videos and also across the individual streams, are registered in each of the two scenarios under examination 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] AccuStream iMedia Research, Inc. [Online]. Available: http://www.accustreamresearch.com

[2] Nielsen//NetRatings, Inc. [Online]. Available: http://www.nielsen-netratings.com/

[3] N. Bjork and C. Christopoulos, "Transcoder architectures for video coding," *IEEE Trans. Consum. Electron.*, vol. 44, pp. 88–98, Feb. 1998.

[4] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," *IEEE Trans. Multimedia*, vol. 2, no. 2, pp. 101–110, Jun. 2000.

[5] T. Warabino, S. Ota, D. Morikawa, M. Ohashi, H. Nakamura, H. Iwashita, and F. Watanabe, "Video transcoding proxy for 3G wireless mobile internet access," *IEEE Commun. Mag.*, vol. 38, no. 10, pp. 66–71, Oct. 2000.

[6] J. Xin, C.-W. Lin, and M.-T. Sun, "Digital video transcoding," in *Proc. IEEE*, vol. 93, Jan. 2005, pp. 84–97.

[7] 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. Circuits Syst. Video Technol.*, vol. 10, no. 8, pp. 1374–1387, Dec. 2000.

[8] M. Khansari, A. Zakauddin, W.-Y. Chan, E. Dubois, and P. Mermelstein, "Approaches to layered coding for dual-rate wireless video transmission," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, vol. 1, Austin, TX, Oct. 1994, pp. 258–262.

[9] S. R. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 6, pp. 983–1001, Aug. 1997.

[10] H. Radha, Y. Chen, K. Parthasarathy, and R. Cohen, "Scalable Internet video using MPEG-4," *Signal Process.: Image Commun.*, vol. 15, no. 1-2, pp. 95–126, Sep. 1999.

[11] U. Horn, K. Stuhlmüller, M. Link, and B. Girod, "Robust Internet video transmission based on scalable coding and unequal error protection," *Signal Process.: Image Commun.*, vol. 15, no. 1-2, pp. 77–94, Sep. 1999.

[12] J. Chakareski, J. Apostolopoulos, S. Wee, W.-T. Tan, and B. Girod, "R-D hint tracks for low-complexity R-D optimized video streaming," in *Proc.IEEE Int. Conf. Multimedia and Exh. (ICME)*, vol. 2. Taipei, Taiwan, R.O.C., Jun. 2004, pp. 1387–1390.

[13] M. Etoh and T. Yoshimura, "Advances in wireless video delivery," in *Proc. IEEE*, vol. 93, Jan. 2005, pp. 111–122.

[14] 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. IEEE Global Telecomunications Conference (Globecom)*, vol. 5, Dallas, TX, Nov./Dec. 2004, pp. 3027–3031.

[15] 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.

[16] 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 Syst. Video Technol.*, vol. 12, no. 6, pp. 524–534, Jun. 2002.

[17] 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. Consum. Electron.*, vol. 50, no. 1, pp. 158–167, Feb. 2004.

[18] Y. Chen, J. Ye, C. Floriach, and K. Challapali, "Video streaming over wireless LAN with efficient scalable coding and prioritized adaptive transmission," in *Proc. ICIP*, vol. 3, Barcelona, Catalonia, Spain, Sep. 2003, pp. 285–288.

[19] X. Xu, M. van der Schaar, S. Krishnamachari, S. Choi, and Y. Wang, "Fine-granular-scalability video streaming over wireless LAN's using cross layer error control," in *Proc. ICASSP*, vol. 5, Montreal, QC, Canada, May 2004, pp. 989–992.

[20] W. Tu, W. Kellerer, and E. Steinbach, "Rate-distortion optimized video frame dropping on active network nodes," in *Proc. Int. Packet Video Workshop* Irvine, CA, Dec. 2004.

[21] J. Chakareski and P. Frossard, "Rate-distortion optimized bandwidth adaptation for distributed media delivery," in *Proc. ICME*, Amsterdam, The Netherlands, July 2005.

[22] K. Stuhlmüller, N. Färber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1012–1032, Jun. 2000.

[23] J. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," in *Proc. SPIE Conf. on Visual Communications and Image Processing*, vol. 4310, San Jose, CA, Jan. 2001, pp. 392–409.

[24] E. Masala and J. de Martin, "Analysis-by-synthesis distortion computation for rate-distortion optimized multimedia streaming," in *Proc. ICME*, vol. 3, Baltimore, MD, Jul. 2003, pp. 345–348.

[25] J. Chakareski, J. Apostolopoulos, W.-T. Tan, S. Wee, and B. Girod, "Distortion chains for predicting the video distortion for general packet loss patterns," in *Proc. ICASSP*, vol. 5. Montreal, QC, Canada, May 2004, pp. 1001–1004.

[26] D. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Nashua, NH: Athena Scientific, May 1998.

[27] L. He and J. Walrand, "Pricing and revenue sharing strategies for internet service providers," in *Proc. INFOCOM*, Miami, FL, Mar. 2005.

[28] K. Pahlavan and P. Krishnamurthy, *Principles of Wireless Networks: A Unified Approach*, 1st ed. Englewood Cliffs, NJ: Prentice-Hall, Dec. 2001.

[29] Synergy Research Group, Inc. [Online]. Available: http://www.srgresearch.com

[30] A. Sicher, "HiperLAN/2 and the evolution of wireless LAN's," in *IEEE Symp. on Emerging Technologies: Broadband, Wireless Internet Access*. Richardson, TX, Apr. 2000.

[31] A. Davies, "An overview of bluetooth wireless technology and some competing LAN standards," in *Proc. Int. Conf. Circuits and Systems for Communications*. St Petersberg, Russia, Jun. 2002, pp. 206–211.

[32] IEEE 802.11 [Online]. Available: http://grouper.ieee.org/groups/802/11/

[33] J. Chakareski, "Distributed media cooperation for enhanced video communication," in *Proc. Int. Packet Video Workshop*, Hangzhou, China, Apr. 2006.

[34] *Telecom. Standardization Sector of ITU, Video Coding for Low Bitrate Communication*, Mar. 2003. Draft ITU-T Rec. H.264.

[35] W. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*. Boston, MA: Addison-Wesley, 1994, vol. 1.

[36] M. Kalman, B. Girod, and P. van Beek, "Optimized transcoding rate selection and packet scheduling for transmitting multiple video streams over a shared channel," in *Proc. ICIP*, Genova, Italy, Sept. 2005.

[37] P. van Beek, H. Pan, I. Sezan, and S. Deshpande, "Adaptive streaming of high-quality video over wireless LANs," in *Proc. SPIE Conf. Visual Communications and Image Processing*. San Jose, CA, Jan. 2004.

Jacob Chakareski received the B.S. degree from Ss. Cyril and Methodius University, Skopje, Macedonia, in 1996, the M.S. degree from Worcester Polytechnic Institute, Worcester, MA, in 1999, and the Ph.D. degree from Rice University, Houston, TX, in 2005, all three in electrical engineering. He performed his doctoral thesis research at Stanford University, Palo Alto, CA from 2002 to 2005.

He currently is a Postdoctoral Researcher at the Swiss Federal Institute of Technology (EPFL), Lausanne. He has held industrial research positions with Microsoft and Hewlett-Packard. Dr. Chakareski has co-authored over 50 international publications and has two pending patent applications. His research interests include networked media systems, statistical signal processing, communication theory and Macedonian history.

Dr. Chakareski received the best student paper award at the VCIP 2004 conference.

Pascal Frossard (S'96–M'01–SM'04) received the M.S. and Ph.D. degrees, both in electrical engineering, from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 1997 and 2000, respectively. In 1998, he graduated from the Graduate School of the Communications System Division of the EPFL.

From 1998 to 2000, he worked with the Signal Processing Laboratory, EPFL, as a research and teaching assistant under a grant from Hewlett-Packard. Between 2001 and 2003, he was a member of the research staff at the IBM T. J. Watson Research Center, Yorktown Heights, NY, where he worked on media compression and streaming technologies. Since April 2003, he has been an assistant professor at EPFL. His research interests include image representation and coding, nonlinear representations, visual information analysis, joint source and channel coding, multimedia communications, and multimedia content distribution. He works on nonlinear signal transforms applied to visual information, and adaptive media streaming algorithms.

Dr. Frossard has been the General Chair of IEEE ICME 2002 (Lausanne, Switzerland), and member of the organizing or technical program committees of numerous conferences. He has served as Guest Editor of special issues on Streaming Media (IEEE TRANSACTIONS ON MULTIMEDIA), on Media and Communication Applications on General Purpose Processors: Hardware and Software Issues (*Journal of VLSI SPSS*), and on Image and Video Coding Beyond Standards (*Journal of Signal Processing*). He is an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA (2004–) and of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2006–), and he served as a member of the Editorial Board of the *EURASIP Journal of Signal Processing* (2003–2005). Since 2004, he has served as Vice-Chair of the IEEE Multimedia Communications Technical Committee, as a member of the IEEE Multimedia Signal Processing Technical Committee, and of the IEEE Multimedia Systems and Applications Technical Committee. He received the Swiss NSF Professorship Award in 2003, and the IBM Faculty Award in 2005