Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Distributed media cooperation for enhanced video communication

CHAKARESKI Jacob

(*Ecole Polytechnique Fédérale de Lausanne, Signal Processing Institute, LTS4, Lausanne CH-1015, Switzerland*)

E-mail: jakov.cakareski@epfl.ch

Received Dec. 2, 2005; revision accepted Feb. 17, 2006

Abstract: The author designed two algorithms for distributed cooperation among multiple video streams sharing common communication resources. The algorithms take advantage of an optimization framework that characterizes video packets such that joint resource allocation can be implemented not only over the packets of a single stream, but also across packets of different streams. The first algorithm enables collaboration among multiple video senders in an 802.11 CSMA/CA wireless network such that their joint performance is maximized. Via the algorithm, the users cooperatively establish transmission priorities based on the assigned characterizations of their video packets. The second technique allows for low-complexity joint bandwidth adaptation of multiple video streams at intermediate network nodes in the Internet in order to maximize the overall network performance. The author analyzes the advantages of the proposed algorithms over conventional solutions employed in such scenarios. It is shown that depending on system parameters such as available network data rate the proposed techniques can provide substantial gains in end-to-end performance.

Key words: Media cooperation, Distributed video streaming, Wireless networks, Resource allocation, Rate-quality optimization
doi:10.1631/jzus.2006.A0773 **Document code:** A **CLC number:** TN919.8

INTRODUCTION

According to yearly statistics the demand for multimedia traffic over computer networks is constantly on the increase (Nielsen//NetRatings, http://www.nielsen-netratings.com/). Therefore, it is inevitable that situations where different media streams will need to share communication resources will become commonplace. In such cases, it would be of interest to optimize the overall performance of the network. In other words, resources should be allocated across the competing media streams such that their aggregate end-to-end performance is maximized, for the given resources.

The present paper addresses this issue in two scenarios. The first one is wireless streaming over 802.11 CSMA/CA networks where competing media streams share the wireless channel (Pahlavan and Krishnamurthy, 2001). An algorithm is proposed that enables the video senders to poll themselves in terms of transmission order, collaboratively and without involvement from the access point. Transmission priorities are established based on rate-quality characterizations assigned to every packet ahead of time. This results in optimal allocation of channel time to the individual user that maximizes their overall performance.

In the second scenario under consideration, multiple incoming video flows contend for resources at an intermediate network node (queue) in the Internet. An algorithm was designed that provides for low-complexity joint bandwidth adaptation of the video flows at the node such that their aggregate performance is optimized. The adaptation of the flows is again based on rate-quality characterizations with which every video packet is tagged at its original sender. These are obtained using an optimization framework for pruning a video source at different data rates. The two communication scenarios examined in this paper are illustrated in Fig.1.

Prior work related to the present paper can be classified into two groups. The first one consists of contemporaneous works on wireless streaming in local area networks (LANs). In particular, Bucciol *et*

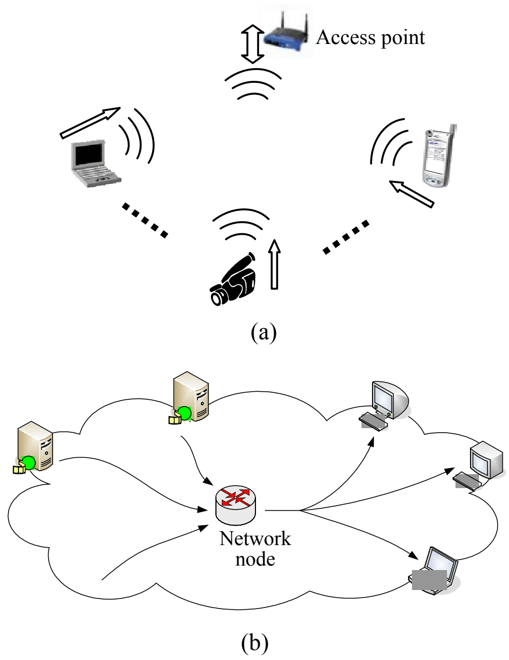

Fig.1 Sharing resources among media streams (a) on a wireless link; (b) at an intermediate network node

al.(2004) proposed a cross-layer Automatic Repeat reQuest (ARQ) strategy for video streaming in 802.11 wireless LANs which gives priority to perceptually more important packets at (re)transmission. In (Li and van der Schaar, 2004), a transmission strategy is examined that provides adaptive quality-of-service (QoS) to layered video for streaming over 802.11 WLANs. However, no rate-distortion optimization is performed. Similarly, in (Majumdar *et al*., 2002; Chen and Wei, 2004), hybrid transmission techniques that combine ARQ and Forward Error Correction (FEC) are proposed for improved real-time video transport over WLANs. Furthermore, Chen *et al*.(2003) proposed a system that combines rate-distortion optimized data partitioning and prioritized adaptive (re)transmission for robust streaming of a single video source over a wireless LAN. Likewise, Xu *et al*.(2004) introduced 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.

However, it should be noted that none of these works considers simultaneous transmission of multiple video streams and many of them do not optimize their proposed strategies. It is only recently that joint resource allocation over multiple streams have been considered (Choi *et al*., 2004; Kalman *et al*., 2005; Chakareski and Frossard, 2005c). In particular, Chakareski and Frossard (2005c) proposed an optimization framework for distributed streaming in WLANs, where the access point employs a TDMA scheme to poll the wireless stations in terms of transmission order according to the importance of their individual packets. Similarly, Kalman *et al*.(2005) considered channel time allocation among multiple video streams in a WLAN employing a simple model (Stuhlmüller *et al*., 2000) to approximate the rate-distortion characteristics of the compressed video streams. This in turn allows for closed form solution of the optimization problem under consideration. Finally, Choi *et al*.(2004) examined cross-layer optimization for multi-user video streaming in wireless cellular environments. A base station is equipped with an optimizer that simultaneously decides the choice of radio link layer and application layer parameters associated with the video streams sent from the base station. In contrast to these works, the present paper considers a distributed content based approach for resource sharing, i.e., streaming over an 802.11 CSMA/CA WLAN (IEEE802.11, http://grouper.ieee.org/groups/802/11). Moreover, a technique different from the approach taken in (Chakareski and Frossard, 2005c) is used to characterize the importance of the video packets.

The second group of related work comprises papers that examine bandwidth adaptation at active network nodes. This is a commonly encountered scenario in the Internet today, and it occurs whenever the data rate on the incoming link at a network node exceeds the data rate on the outgoing link. Then, the node needs to decide to drop some of the incoming packets in order to account for the mismatch between incoming and outgoing rates. Keller *et al*.(2000) proposed a strategy for dropping packets from a single incoming video stream that is encoded using the wavelet transform. Reduction in data rate is achieved either by dropping whole video frames, thereby reducing the temporal frame rate of the video, or by preferentially dropping packets carrying higher frequency bands of the encoded frames. No rate-distortion optimization is performed. Balakrishnan and Ramakrishnan (2000), and Bai and Ito (2004) studied bandwidth adaptation via packet dropping for

MPEG-2 encoded video and propose dropping strategies which in essence place different delivery priorities on the different frame types of the encoded video: I, P and B. Only a single video stream is considered and no rate-distortion optimization is performed. Bouazizi (2003) considered rate-distortion optimized packet dropping in the context of proxy-caching for broadcasting of an MPEG-4 encoded single video stream.

Contrary to the works described above, the present paper considers joint bandwidth adaptation of multiple video streams in a rate-quality optimized way. In this regard, our work is most closely related to (Tu *et al*., 2004; Chakareski and Frossard, 2005a) which also study bandwidth adaptation across multiple streams, however with a different technique for characterizing the media packets. As shown in (Chakareski and Frossard, 2005b), this alternative technique is outperformed by the characterization solution employed in our paper, due to its greedy approach.

The rest of the paper is organized as follows. First, we describe the optimization technique for rate-quality characterization of packetized media streams in Section 2. Then, in Section 3 we present the algorithm for distributed collaboration among users in a wireless network that employs these characterizations to enable optimal channel sharing among the users. Section 4 describes next the proposed algorithm for joint bandwidth adaptation of multiple video streams at a network node. Furthermore, in Section 5 we examine the performance of these two algorithms and compare it to that of conventional techniques employed in the scenarios under consideration. Finally, concluding remarks are provided in Section 6.

SOURCE CHARACTERIZATION VIA PRUNING

To characterize the packets of a video stream in terms of their importance for the reconstruction quality of the stream, we employ a technique that was proposed recently in (Chakareski and Frossard, 2005b). In the following, we first briefly describe the principles of the technique and refer the reader to the cited reference for further details. Then, we explain how we have extended this technique for the purposes of joint resource allocation across multiple video streams, which is the focus of the present paper.

Let $R_1, \ldots, R_N$ be a sequence of N monotonically increasing data rates. Using the technique from (Chakareski and Frossard, 2005b), we can classify the packets from a video stream into N sets $S_1, \ldots, S_N$, where the sets S_i are obtained by pruning (dropping packets from) the video source such that the data rate of the pruned source does not exceed the corresponding rates R_i, for $i=1, \ldots, N$.

While reducing the rate of the video source to R_i the algorithm chooses to discard those data units (here we interchangeably use the terms “packet” and “data unit”) from the packetized representation of the compressed source that will contribute to the smallest reduction in video quality. Let the resulting video quality for the pruned representation S_i be denoted as Q_i. It is important to note that the pruning algorithm typically creates embedded sets, i.e., for any two sets S_i and S_j such $i<j$, it holds that $S_i \subset S_j$. So in summary, if S corresponds to the full set of data units of the packetized source, then holds that $S_1 \subset S_2 \subset \ldots \subset S_N \subset S$.

In our case, the difference $R_N - R_1$ is selected such that it covers a wide range of data rates to which the video source could be adapted, for example $R_N/R_1 = 1.5$. Moreover, a sufficiently large N is chosen so that there is fine (incremental) division of the data rate range $R_N - R_1$. Finally, we select R_N such that it corresponds to the encoding rate of the source, i.e., S_N contains all the packets from the video stream ($S_N = S$).

Now, given the preprocessing of the source outlined above we can synthesize its operational rate-quality function using linear interpolation between the points $[R_i, Q_i]$, as shown in Fig.2.

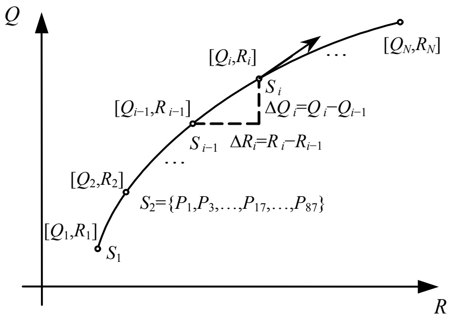

Fig.2 Operational *R*-*Q* function for a video source

The incremental increase in quality and rate that each new set S_i provides relative to its predecessor

S_{i-1} can be determined as $\Delta Q_i=Q_i-Q_{i-1}$, $\Delta R_i=R_i-R_{i-1}$ (for the purposes of our analysis, we can adopt that $R_0=Q_0=0$). Then, to each segment i of the operational R-Q function, we can assign a "gradient" of the function (on that segment) defined as $\lambda_i=\Delta Q_i/\Delta R_i$. In plain words, λ_i denotes the per unit rate increase in quality that adding the segment i will provide to the video source.

Finally, we label each packet $P_j\in S$, for $j=1, 2, \ldots$, with the index i of the set S_i where it appears first, that is, $i=\min_n n$, s.t. $P_j\in S_n$, $n=1, \ldots, N$. For example, the set S_2 may comprise packets P_1, P_3, P_{17}, and P_{87}, among others, as illustrated in Fig.2. In addition, we also label packet P_j with the corresponding gradient λ_i.

The reasons why we selected these two parameters for characterizing the video packets are as follows. If there is only one source that is exploiting a communication resource (say the data rate on a communication link), then employing the set labels i assigned to every packet would be sufficient to allocate the resource over the packets. However, when multiple sources compete for the same resource then we need a mechanism to compare which of the sources provides the highest benefit (increase in quality) per unit of allocated resource (data rate). To this end, we take advantage of the gradients λ_i. These issues will become clear in the next two sections, where we study concrete scenarios in which these packet characterizations are employed.

DISTRIBUTED COLLABORATION IN WLANS

Consider that there are K users in a wireless LAN transmitting video stream over the shared medium (Fig.1a). The network employs CSMA/CA scheme to allow the users to access the channel. We assume that all users are within a transmission range of the access point (one hop communication) and of each other. That is each user can communicate directly with the access point and a transmission of one user can be heard by the rest of the users.

The proposed algorithm that the users employ to send packets over the air consists of two phases. The first phase is information collecting and allows the users to establish transmission order of the packets that they selected to send next. The second phase consists of the actual transmissions of their respective packets, in the order determined at completion of the first phase. The specifics of the algorithm are as follows.

(1) Each user has a window of video packets considered for transmission. The packets are characterized/labeled ahead of time using the procedure explained in Section 2. The users need to decide on their next packet to send from their transmission windows. That is done by selecting the packet with the largest gradient label λ from each window, respectively by the individual users. This choice is motivated by the fact that such a selected packet belongs to a subset of packets of the video stream that provide the biggest increase (relative to the rest of the packets from the transmission window) in video quality per unit of allocated transmission rate. However, the users still do not know what the best order is for transmitting their most important packets that they just selected. This is where the information collecting phase comes in.

(2) Each user publishes on the channel the λ factor associated with its most important packet. The specific order in which these values are announced is irrelevant and can be determined ahead of time. For example, the users can transmit these factors on the channel according to their seniority, i.e., the duration of time for which they have been present on the channel. To ease notation, we assume here that the users send their gradients in a round-robin fashion according to their index k. Note that ordered transmission is possible because the users know about each others' existences, so it suffices for each user k to wait to hear the announcements of the previous $k-1$ users and then to publish its own λ value on the network.

As the 802.11 standard requires strict set of rules when communicating on the shared channel, the users follow these when making the announcements. In particular, each user waits first a pre-subscribed period of time DIFS during which the channel is idle. Only then, a user will send its λ value in a small payload-less packet prescribed by the standard and known as RTS (Ready to Send). In its header the user embeds the corresponding λ factor. The standard also mandates that an RTS packet is always answered by the respective receiver (typically the access point) with another special header-only packet known as CTS (Clear to Send). Therefore, the announcement by the next user will only be sent after the access point returns a CTS packet for the previous announcement.

The RTS/CTS packet pair is provided by the 802.11 standard (IEEE 802.11, http://grouper.ieee.org/groups/802/11/) to allow a station (user) to request a reservation of the channel (RTS) and to receive a confirmation that the reservation has been granted (CTS). The dynamics of the information collecting phase is illustrated in Fig.3, where SIFS is another period of time (between receiving the RTS frame and sending the corresponding CTS frames) prescribed by the standard.

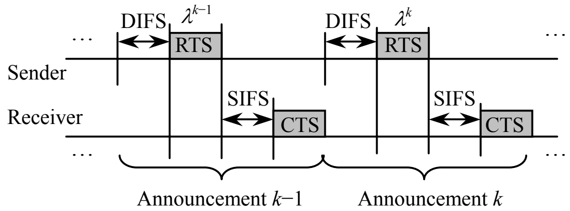

Fig.3 Information collecting phase

At the end of the information collecting phase every user acquires a vector of gradient values, [λ^1, …, λ^K], describing the importance of the next packet that each user is interested in sending, respectively. The users then sort this vector in decreasing order and transmit their respective packets in that order. In particular, let $\boldsymbol{I}$ denote the vector of user indices that correspond to the sorted gradient vector. Then, user $\boldsymbol{I}(k)$ waits to hear the transmissions of the previous k−1 users, $\boldsymbol{I}(1),\ldots,\boldsymbol{I}(k-1)$, before it transmits its own packet. Note that sending the packets in such ordering maximizes the overall network utility in terms of video quality, as more important packets (larger λ factors) are given priority in terms of transmission. This increases the likelihood that these packets will arrive earlier at their respective destinations and therefore will not miss their delivery deadlines (this is the time instance at which a packet is due to be decoded and displayed by the client application at the receiver).

Fig.4 illustrates the data transmission phase at a user. First, the user sends an RTS packet announcing the duration of time for which it needs to get hold of the channel. Followed by a CTS packet from the access point, the user then transmits the actual video packet, which in turn is followed by an acknowledgement packet from the access point, as shown in Fig.4. Then, the next user in the sorted list $\boldsymbol{I}$ can begin the same

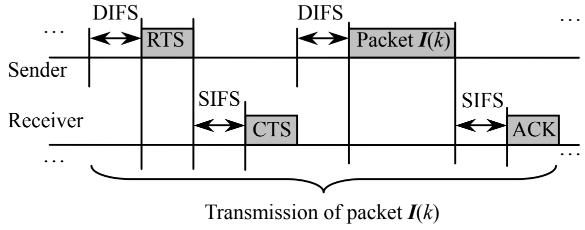

Fig.4 Data transmission phase

procedure to transmit its own packet. Note that it would be more efficient to perform the data transmission phase without the RTS/CTS handshaking, as the channel reservation time for each user can be announced together with the corresponding gradient during the information collecting phase. However, to the best of our knowledge, switching on/off the RTS/CTS mode on a per packet/transmission basis is not provided at present by the 802.11 standard.

After the two phases of the transmission protocol described heretofore is complete, the same procedure is repeated till the users send all their packets. After each cycle of information collecting and data transmission, the users update their transmission windows accordingly. This is done by removing from a window packets that have been already transmitted or have expired in the meantime (their delivery deadlines have already passed). In addition, new packets that have not been considered for transmission previously are simultaneously added to the window. When a user does not have any more packets to consider for transmission, i.e., its transmission window becomes empty, it publishes a gradient λ=0. This will signal to the other users that the user is not interested in transmission any longer, so it can be skipped during the data transmission phase.

JOINT BANDWIDTH ADAPTATION

Here, we examine the scenario where there are K media streams arriving at a network node (Fig.1b). In addition to the media streams there is also cross traffic of packets from other flows sharing the node's resources (forwarding data rate, buffer size, CPU processing time, etc.) with the media packets. Consider that the effective (the actual data rate on the link minus the average rate of the cross-traffic) data rate on the outgoing link that the node can allocate to the K

streams is R. The node is interested in partitioning R across the streams such that their aggregate performance is maximized.

The problem under consideration can be formalized as follows. Let R^k, k=1, …, K, be the rates allocated to the individual streams, and let $Q^k(R^k)$ be the corresponding video quality of each stream. This is a classical resource allocation problem where the node is interested in finding the rate vector $(R^1, \ldots, R^K)^*$ that maximizes the overall quality $\sum_{k=1}^{K} Q^k(R^k)$, such that $\sum_{k=1}^{K} R^k \le R$. In other words, $(R^1,\ldots,R^K)^*$=arg min $\sum_{k=1}^{K} Q^k(R^k)$, s.t. $\sum_{k=1}^{K} R^k \le R$.

Using the method of Lagrange multipliers (Everett, 1963; Shoham and Gersho, 1988; Chou *et al*., 1989), the constrained optimization problem from above can be reformulated as non-constrained optimization, where

$$(R^1,\ldots,R^K)^* = \arg\min \sum_{k=1}^{K} Q^k(R^k) + \lambda\left(\sum_{k=1}^{K} R^k - R\right),$$

for a suitably selected Lagrange multiplier $\lambda>0$.

Optimization problems of this nature have been treated extensively in the past, in areas such as video compression, communications and operations research (Ortega and Ramchandran, 1998; Sullivan and Wiegand, 1998). The novel contribution of the present paper is that the problem under consideration can be quickly solved by the node by exploiting the labelling of the video packets that has been applied to every stream ahead of time, as described in Section 2.

In particular, when the rate-quality functions $Q^k(R^k)$ are differentiable the solution to the non-constrained formulation of the problem reduces to solving $\mathrm{d}Q^k(R^k)/\mathrm{d}R^k=\lambda$ independently, for k=1, …, K. Therefore, in the scenario considered here, the network node only needs to find the largest gradient $\lambda^k \in \{\lambda_1^k,\ldots,\lambda_{Nk}^k\}$ (the list of gradient values for every stream can be made available to the node via a preamble packet sent ahead of the steam) for every stream k, such that $\lambda^k \le \lambda$. Note that this provides the best approximation to $\mathrm{d}Q^k(R^k)/\mathrm{d}R^k=\lambda$ that does not exceed the overall budget R. Moreover, given a sufficiently large N_k for every stream, the approximation may approach closely the continuous case solution $\sum_{k=1}^{K} R^{k*}=R$.

Finally, once the node has determined the gradients for every stream that satisfy the above condition, adaptation of the streams is done in a straightforward manner. Specifically, let $[\lambda^1, \ldots, \lambda^K]^*$ be the vector of maximum gradient values for every stream such that $\lambda^{k*} \le \lambda$, for k=1, …, K. Then, adaptation of stream k is performed by simply dropping all packets from stream k that are marked with a gradient value smaller than λ^{k*}. In other words, appropriate action for every packet in stream k is performed by simply filtering (to drop or to pass) the packet based on its tagged gradient against the common threshold λ^{k*}. Hence, in this way low-complexity adaptation of the streams is achieved that nonetheless exploits the available resource in the most efficient manner.

EXPERIMENTAL RESULTS

This section investigates the performance of the two algorithms for distributed media collaboration proposed in the present paper. The video content that is employed in the experiments consists of four standard test video sequences: Foreman, Mother & Daughter, Carphone, and Salesman. The size of the video frames is QCIF. Using JM 2.1 of the JVT/H.264 video compression standard (Telecom. Standardization Sector of ITU, 2003) the sequences have been encoded at 10 fps and an average luminance (Y) $PSNR$ of about 36 dB. The size of a Group of Pictures (GOP) is 20 frames, comprising an I frame followed by 19 consecutive P frames. The individual rate/quality encoding characteristics of the four sequences are shown in Table 1.

Table 1 Encoding characteristics of the four sequences

Sequence	Rate (kbps)	*Y-PSNR* (dB)
Foreman	82.23	35.75
Carphone	80.94	36.88
Mother & Daughter	38.65	36.77
Salesman	40.09	36.23

Each sequence has been processed using the packet classification framework from Section 2. It should be pointed out that the video content Mother &

Daughter and Salesman is characterized in general with a steeper slope of their respective *R*-*Q* functions relative to the content represented with the sequences Foreman and Carphone, as established through visual inspection. This means that the former video content would typically be treated preferentially in a joint resource allocation procedure, as explained earlier. The results of the analysis presented in the next two sections are in agreement with this conclusion.

In the experiments that follow, performance of an algorithm is measured in terms of the average *Y-PSNR* of the frames of a reconstructed video content at a receiver, averaged over all receivers. In addition, we also examine the individual performances at each receiver. Frames that are not delivered on time at a receiver are replaced using previous frame error concealment. In the discussion and the corresponding experimental results, the systems that employ the proposed algorithms are denoted PackClass. Their performances are examined against reference systems denoted Baseline which represent conventional solutions employed at present in the scenarios under consideration.

Wireless streaming

There are four users in this setting, represented by the four video sequences respectively, attempting to send their video content over the wireless channel. The play-out delay (*InitialDelay*) for the video content at their respective receivers is set to 200 ms. There are two systems under examination here. As introduced earlier, the system PackClass employs the algorithm from Section 6 to coordinate the transmissions of the users. The competing system, denoted Baseline, employs the conventional Distributed Coordination Function (DCF) to allow the users to share the channel, as prescribed by the 802.11 CSMA/CS standard. Users in this system choose to transmit packets according to their delivery deadlines. In particular, the video packet with the earliest deadline from a transmission window is selected to be sent next. Note that in the Baseline system the users perform their transmissions without reference to the importance of the individual video packets in a rate-quality sense.

In Fig.5, we show the performances of the two systems as a function of the available data rate on the wireless link, normalized with the aggregate encoding rate for the four streams. Normalized data rates in the range 0.4~1.5 are considered. It can be seen that PackClass outperforms Baseline with a significant margin for data rates less than one. For example, at normalized data rate of 0.8, the improvement in performance relative to the Baseline system is around 8 dB. In addition, the gains in performance increase as the data rate decreases. The improved performance is due to the fact that PackClass takes advantage of the importance of the individual packets for the reconstruction quality of a stream when scheduling their transmission. Moreover, by trading data rate across packets of different streams jointly, PackClass is able to provide the best possible overall performance for the given available data rate.

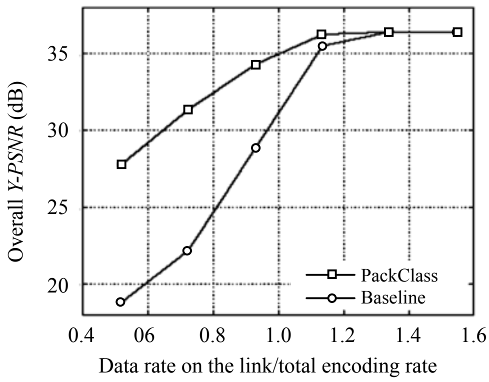

Fig.5 Distributed streaming: Overall performance as a function of available data rate on the wireless link (Average quality over 4 users: *InitialDelay*=200 ms)

On the other hand, when the data rate is increased to above one, we can see from Fig.5 that the performances of the two systems converge. This is also expected, as here there is sufficient bandwidth to transmit and deliver on time (almost) all of the video packets. Hence, the advantages of knowing their importance and allocating resources jointly across them become irrelevant. Note from Fig.5 that data rate above 1.2 is needed for the two systems to reach full performance, as extra rate is needed to account for the peak data rate requirements of each stream and to compensate for the various overheads introduced by the communication protocols.

Next, we examine the performances of every user. These are shown in Fig.6 showing that also over the individual users PackClass provides significant improvements relative to Baseline. As we observed previously, the gains in performance are most pronounced for data rates below one. For example, at data rate=0.8, Baseline is outperformed with margins of roughly 10, 9, 6, and 4.5 dB, for Foreman, Mother & Daughter, Carphone, and Salesman, respectively, as seen from Fig.6. The gains in performance are quite significant and are due to the fact that the optimized system exploits the importance of information associated with the individual packets, when performing scheduling decisions, as explained earlier.

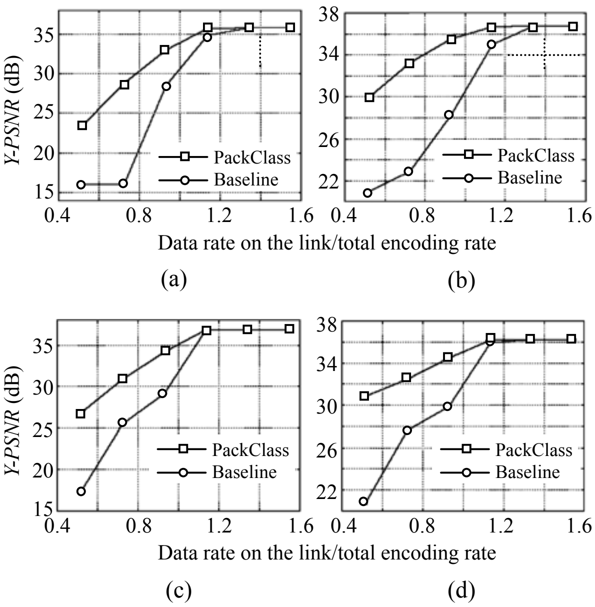

Fig.6 Distributed streaming: Individual performance of every user as a function of available data rate on the wireless link (Video quality over all users: *InitialDelay*=200 ms). (a) Foreman; (b) Mother & Daughter; (c) Carphone; (d) Salesman

Finally, note that the performance of PackClass for Mother & Daughter and Salesman scales more gracefully with the data rate relative to the respective performances for Foreman and Carphone. This is because PackClass preferentially allocates rate across packets of the former two sequences relative to packets of the latter two sequences. This in turn is due to the higher utility that the video content represented by Mother & Daughter and Salesman provides per unit allocated rate, as discussed earlier in the context of their operational *Q*-*R* functions at the beginning of Section 5. On the other hand, Baseline treats the packets of all users (sequences) equally, and hence the roughly equal degradation in quality for each one of them, as the data rate is decreased.

Network node adaptation

This section considers the scenario where there are four video flows arriving at an intermediate network node. Based on the available forwarding data rate at the node relative to the aggregate data rate of incoming packets, the node may need to adapt the flows to account for the mismatch in rates. That is done by dropping incoming packets from the flows. To this end, three different bandwidth adaptation algorithms are employed by the node and tested for performance. PackClass denotes the algorithm for distributed collaboration among media streams proposed in Section 4. Baseline performs allocation of the forwarding data rate to the incoming flows in proportion to their incoming (encoding) rates, and without regard for the importance of the individual packets, as explained earlier.

Finally, the third algorithm under investigation is denoted NumDesc and places dropping priorities on the incoming packets according to the number of their descendants. In particular, in predictive video coding a frame is typically encoded with a reference to a previous frame. This creates dependency chains between the frames. For example, to decode the last frame in a GOP one needs to decode first the rest (or a subset of it in the case of existence of B-frames) of the (previous) frames from the GOP. Thus, in NumDesc I-frames have the highest priority equal to the number of frames in their respective GOPs, while the last frames (including any B-frames) in every GOP have the smallest priority [the argument still holds in case of B-frames, as then streams (GOPs) are packetized in decoding order, i.e., B-frames come after P-frames]. Then, when the incoming data rate of the flows needs to be reduced, packets with the smallest priority are dropped first, followed by packets with the second to the last priority, etc., till the incoming and outgoing data rates are brought in accord. Packets with equal priority are treated uniformly in NumDesc. That is, if a number (but not all) of equal priority packets needs to be dropped, the selection is made at random.

We decided to include NumDesc in the investigation here, in order to provide us with a better understanding of the performance of PackClass. In particular, since the two systems treat video packets preferentially, we are interested in finding out how much further improvement can be provided by taking into account the importance of the video packets as done in PackClass, relative to the case when that importance is simply summed up in the number of descendant frames associated with a video frame.

In Fig.7, we examine the performance of the three algorithms as a function of the forwarding data rate at the node, expressed in percent of the aggregate data rate of the incoming flows. It can be seen that when no flow adaptation needs to be performed (100% forwarding rate), the three systems under investigation perform alike, which is expected. However, as the data rate of the incoming flows needs to be reduced, we can see from Fig.7 that the performance of the Baseline system rapidly deteriorates. For example, at forwarding data rate of 80% the performance of Baseline has reduced to roughly 28 dB, from a 36 dB performance at full rate. On the other hand, the performance of PackClass has degraded gradually for only 2 dB in the same comparison. Furthermore, even when additional data rate reduction needs to be performed PackClass still maintains a commendable video quality, as illustrated by the achieved 30 dB overall performance at 50% rate reduction relative to only 24 dB for Baseline at forwarding data rate of 60% (Fig.7).

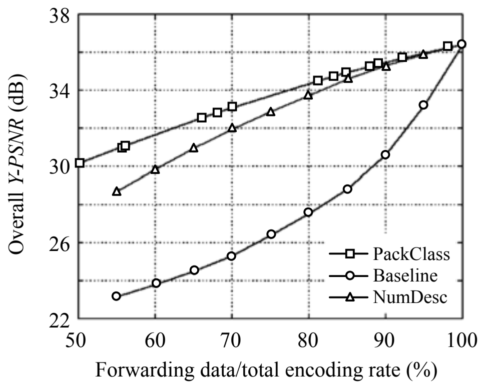

Fig.7 Bandwidth adaptation: Overall performance as a function of forwarding data rate (%) (Average quality for joint adaptation of 4 users)

The reasons for the differences in performance between systems such as PackClass and Baseline when performing allocation of insufficient resources were explained in Section 5.1 in the context of wireless streaming. In essence, PackClass takes advantage of the importance information associated with the video packets to optimize the allocation of rates to the individual flows, as explained earlier.

Now, it is interesting that NumDesc provides the same performance to PackClass for small rate reductions. This is intuitive because the least important frames in a video sequence are typically those found at the end of GOPs. So for small rate reductions, it simply does not provide a lot of difference in video quality if the very last frames across different GOPs and sequences are treated differentially. However, as more frames need to be dropped since the incoming data rate needs to be reduced further, it makes sense to provide some mechanism of differentiation across different GOPs and video streams, of frames at same locations in their respective GOPs. This argument is supported by the increasing difference in performance between PackClass and NumDesc as the forwarding data rate reduces, as illustrated in Fig.7. For example, at 70% forwarding data rate there is already a margin of 1 dB in performance between the two systems, which increases further to 2 dB for a data rate reduction of 45%, as seen from Fig.7.

Finally, we examine the performances of PackClass, Baseline and NumDesc over the individual video flows. These are shown in Fig.8. It can be seen that as in the case of wireless streaming PackClass outperforms Baseline with a significant margin also for the individual video flows. For example, for a data

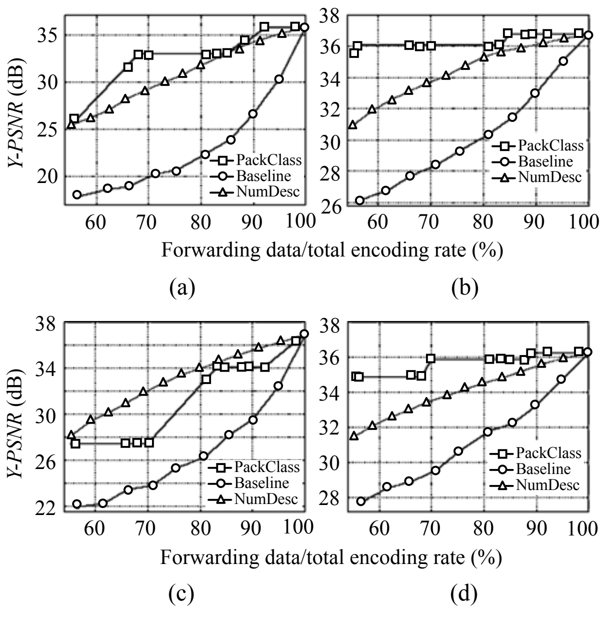

Fig.8 Bandwidth adaptation: Individual performances of the video flows as a function of forwarding data rate (%). (a) Foreman; (b) Mother & Daughter; (c) Carphone; (d) Salesman

rate reduction of 10%, improvements in performance of around 9, 4, 4.5, and 3 dB, are registered respectively in the case of Foreman, Mother & Daughter, Carphone, and Salesman, as seen from Fig.8.

In addition, note the more gradual degradation in performance as the forwarding data rate is reduced for Mother & Daughter and Salesman, relative to the performances for the other two video flows, in the case of PackClass. This was also observed earlier in Section 5.1. It is interesting to note that because of this behaviour PackClass outperforms NumDesc much more significantly in the case of Mother & Daughter and Salesman relative to the cases for the other two flows. In fact, PackClass underperforms relative to NumDesc in the case of Carphone, as shown in Fig.8c. Nonetheless, by taking advantage of the different operational R-Q functions for the individual video flows that enable it to trade-off data rate across the packets of the flows in the most efficient manner, PackClass is still able to maintain superior overall performance over all data rates under consideration.

CONCLUSION

We have presented two algorithms for distributed collaboration among multiple video streams. The algorithms rely on an optimization framework for labelling the video packets of a stream in terms of their importance for the reconstruction quality of the stream, as a function of the available data rate. This characterization enables the two algorithms to trade-off quality vs data rate not only within a stream, but also across multiple streams such that the overall performance of the systems where they are deployed is maximized. In the context of wireless streaming over 802.11 CSMA/CA networks, we proposed an algorithm that allows the users to cooperate distributively in sharing the wireless channel towards the goal of optimizing the overall performance of the network. In the context of network node packet management, we presented an algorithm for joint bandwidth adaptation of multiple incoming video flows. In both cases, the two algorithms provide significant improvements in performance relative to conventional solutions implemented in practice today.

ACKNOWLEDGEMENT

The author would like to thank John Apostolopoulos of HP Labs for the early discussions on distributed streaming in WLANs and to Xiaoqing Zhu of Stanford University for the useful feedback on the IEEE 802.11b standard.

References

Bai, Y., Ito, M., 2004. Network-Level Loss Control Schemes for Streaming Video. Proc. ICME.

Balakrishnan, R., Ramakrishnan, K., 2000. Active Router Approach for Selective Packet Discard of Streamed MPEG Video under Low Bandwidth Conditions. Proc. ICME.

Bouazizi, I., 2003. Size-Distortion Optimized Proxy Caching for Robust Transmission of MPEG-4 Video. Proc. MIPS.

Bucciol, P., Davini, G., Masala, E., Filippi, E., Martin, J.D., 2004. Crosslayer Perceptual ARQ for H.264 Video Streaming over 802.11 Wireless Networks. Proc. Globecom.

Chakareski, J., Frossard, P., 2005a. Rate-Distortion Optimized Bandwidth Adaptation for Distributed Media Delivery. Proc. ICME.

Chakareski, J., Frossard, P., 2005b. Low-Complexity Adaptive Streaming via Optimized a Priori Media Pruning. Proc. MMSP.

Chakareski, J., Frossard, P., 2005c. Distributed Packet Scheduling of Multiple Video Streams over Shared Communication Resources. Proc. MMSP.

Chen, M., Wei, G., 2004. Multi-stages hybrid ARQ with conditional frame skipping and reference frame selecting scheme for real-time video transport over wireless LAN. *IEEE Trans. Cons. Electronics*, **50**(1):158-167. [doi:10.1109/TCE.2004.1277856]

Chen, Y., Ye, J., Floriach, C., Challapali, K., 2003. Video Streaming over Wireless LAN with Efficient Scalable Coding and Prioritized Adaptive Transmission. Proc. ICIP.

Choi, L.U., Kellerer, W., Steinhach, E., 2004. Cross Layer Optimization for Wireless Multi-User Video Streaming. Proc. ICIP.

Chou, P.A., Lookabaugh, T., Gray, R.M., 1989. Entropy-constrained vector quantization. *IEEE Trans. ASSP*, **37**(1):31-42.

Everett, H., 1963. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, **11**(3):399-417.

Kalman, M., Girod, B., van Beek, P., 2005. Optimized Transcoding Rate Selection and Packet Scheduling for Transmitting Multiple Video Streams over a Shared Channel. Proc. ICIP.

Keller, R., Choi, S., Dasen, M., Decasper, D., Fankhauser, G., Plattner, B., 2000. An Active Router Architecture for Multicast Video Distribution. Proc. INFOCOM.

Li, Q., van der Schaar, M., 2004. Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation. *IEEE Trans. Multimedia*, **6**(2):278-290. [doi:10.1109/TMM.2003.822792]

Majumdar, A., Sachs, D.G., Kozintsev, I.V., Ramchandran, K., Yeung, M.M., 2002. Multicast and unicast real-time video streaming over wireless LANs. *IEEE Trans. CSVT*, **12**(6): 524-534.

Ortega, A., Ramchandran, K., 1998. From rate-distortion theory to commercial image and video compression technology. *IEEE SP Magazine*, **15**(6):20-22. [doi:10.1109/MSP.1998.733494]

Pahlavan, K., Krishnamurthy, P., 2001. Principles of Wireless Networks: A Unified Approach, 1st Ed. Prentice-Hall, Englewood Cliffs, NJ.

Shoham, Y., Gersho, A., 1988. Efficient bit allocation for an arbitrary set of quantizers. *IEEE Trans. ASSP*, **36**(9): 1445-1453.

Stuhlmüller, K., Färber, N., Link, M., Girod, B., 2000. Analysis of video transmission over lossy channels. *IEEE JSAC*, **18**(6):1012-1032.

Sullivan, G.J., Wiegand, T., 1998. Rate-distortion optimization for video compression. *IEEE SP Magazine*, **15**(6):74-90. [doi:10.1109/79.733497]

Tapia, R.A., 2000. Mathematical Optimization and Lagrange Multiplier Theory for Scientists and Engineers. Course notes CAAM-460, Rice University, Houston, TX.

Telecom. Standardization Sector of ITU, 2003. Video Coding for Low Bitrate Communication. Draft ITU-T Recommendation H.264.

Tu, W., Kellerer, W., Steinbach, E., 2004. Rate-Distortion Optimized Video Frame Dropping on Active Network Nodes. Proc. Packet Video Workshop.

Xu, X., van der Schaar, M., Krishnamachari, S., Choi, S., Wang, Y., 2004. Fine-Granular-Scalability Video Streaming over Wireless LANs Using Cross Layer Error Control. Proc. ICASSP.