Quality-oriented Video Transmission with Pipeline Forwarding

Mario Baldi, *Member, IEEE*, Juan Carlos De Martin, *Member, IEEE*, Enrico Masala, *Member, IEEE*,
Andrea Vesco, *Member, IEEE*

***Abstract*—This work proposes a quality-oriented multimedia delivery framework that tackles the issue of optimizing video broadcasting and interactive video applications over packet networks with respect to both resource utilization and user perceived quality. Previous work showed that the quality of service requirements of multimedia applications can be optimally satisfied by pipeline forwarding of packets by keeping delay controlled and resource utilization high, while enabling highly scalable network devices. These properties are key in today networks to enable valuable (i.e., chargeable for) services and to avoid that the traffic increase due to broadband video either collapses existing networks or forces the deployment of high cost, cutting-edge technology to properly upgrade them. However, the current Internet is not based on such technology and its incremental introduction raises questions on how to handle video packets generated by pipeline-forwarding-unaware sources. This work proposes to use the perceptual importance of the carried video samples to determine which packets shall be transferred with pipeline forwarding — thus receiving deterministic service — and which with a traditional, e.g., best effort or differentiated service. Two new scheduling algorithms are proposed, with an extensive analysis and simulation results, which also investigate the impact of the encoding scheme. Performance bounds have been established by comparing the proposed algorithms with an exhaustive-search approach, showing that the performance is within 2 dB PSNR from the optimal solution in the worst case.**

***Index Terms*—Pipeline forwarding, quality of service, multimedia communication, video streaming and broadcasting.**

I. Introduction

VARIOUS multimedia applications, such as video broadcasting (e.g., IPTV), video on demand, telephony over IP, voice over IP (VoIP), are becoming more widely available. These applications are often referred to as real-time to juxtapose them to traditional data applications as timely packet delivery is important for them to work properly.

However, the large-scale development of traditional broadcasting services such as IPTV over packet networks originally designed for generic data applications, such as the current Internet, faces numerous challenges stemming from the stringent QoS and high bandwidth requirements of multimedia applications. Packet networks originally designed for generic data applications are not engineered to tightly control the delay packets experience in routers where they might contend for resources (e.g., transmission capacity), consequently be queued for a variable time, and possibly be dropped. Moreover, multimedia applications are usually of a streaming nature as they generate a more or less continuous flow of data and not elastic, i.e., they need a minimum fraction of their data to reach the destination and do not adapt to particularly poor network service.

Currently the requirements of multimedia applications are commonly satisfied through overprovisioning, i.e., by keeping the network lightly loaded so that contention for network resources is low and queuing time consequently small. This approach is not feasible if multimedia traffic grows faster than technology enables proportionally more powerful network infrastructures. And this might be the case not only because a larger fraction of broadband users might subscribe current multimedia services, but especially because more bandwidth-demanding services such as 3D video on demand, high quality videoconferencing, high definition TV, distributed gaming, (3D) virtual reality, and remote surveillance, might become the dominant traffic sources in the future Internet.

This work presents a quality-oriented multimedia delivery framework which tackles the issue of optimizing multimedia applications in general, and specifically video broadcasting applications, over packet networks with respect to both resource utilization and user perceived quality.

Previous work [1] showed that Pipeline Forwarding (PF) of packets [2] can satisfy the quality of service requirements of multimedia applications while ensuring high network utilization [3]. PF properties stem from network nodes sharing a common time reference (CTR) and hold also when multicasting of packets is performed [4], thus both broadcasting services and group communications can enjoy PF benefits. Moreover, pipeline forwarding enables the realization of highly scalable network nodes [5]. Consequently, PF firstly enables overcoming the scalability limitations of the overprovisioning-based approach by providing efficient support of multimedia applications (i.e., high network utilization). Service providers can thus offer new multimedia services at competitive prices to a large

Manuscript received September 3, 2007. Mario Baldi, Juan Carlos De Martin, Enrico Masala and Andrea Vesco are with Control and Computer Engineering Department, Technical University Of Turin, Corso Duca degli Abruzzi 24, 10129 Turin Italy (e-mail: mario.baldi@polito.it, demartin@polito.it, masala@polito.it, andrea.vesco@polito.it).

customer base without overwhelming the current infrastructure and needing to upgrade it using expensive cutting-edge technology. Secondly, as various analysts, service providers, and equipment vendors are forecasting[1], when current and novel bandwidth intensive multimedia services will get deployed on a wide scale, current network infrastructures will be overwhelmed with huge amounts of traffic. PF is key in enabling the implementation of highly scalable network nodes [5] that will be able to overcome the switching bottleneck that is affecting switching solutions and architectures currently deployed in network devices.

As it will be discussed in Section II, resources that guarantee service quality are reserved in PF nodes in a periodic fashion. As proposed in [1], the reservation period can be made a multiple of the video frame period and different amounts of resources can be reserved for different video frames in the reservation period. [1] presents a PF-aware video codec that keeps the amount of bits used to encode each frame within the reservation. The resulting multimedia system is shown to be optimal in terms of both end-to-end delay and network resource utilization. However, PF is not deployed in the current Internet and the only realistic way to introduce it is incremental. This paper complements [1] by addressing the issue of how to handle video packets generated by PF-unaware sources. The solutions proposed and analyzed in this paper can be applied at either the source itself or the interface between a portion of the network deploying conventional packet scheduling techniques and one implementing PF.

This work proposes to use the perceptual importance of the carried video samples to determine which packets shall use reserved resource — i.e., be transferred with PF, thus receiving deterministic service — and which with a traditional, e.g., best effort or differentiated, service given that the amount of bits encoding a frame is irregularly variable, especially when the video stream is encoded with constant quality. The importance of video samples is computed according to the method proposed in [6], using a low-complexity model-based implementation [7]. This importance estimation method has been already used in perceptually-optimized multimedia communication schemes [8][9]. Preliminary results of the first implemented variant of the solutions analyzed in this work were presented in [10]. Here two new scheduling algorithms are proposed, with extensive analytical analysis and simulation results aimed at also investigating the impact of various encoding schemes characterized by different tradeoffs between encoded video quality and bit rate fluctuations. Performance bounds have been established by comparing the proposed algorithms with an exhaustive-search approach.

The paper is organized as follows. Section II discusses PF by presenting its operating principles and properties. The approach deployed to model perceptual importance associated to video samples and distortion resulting from their loss is presented in Section III. Section IV presents the network model, provides a formulation of the problem of minimizing the expected distortion of the video sequence at the receiver as an optimization problem, analytically models the various packet scheduling options, and introduces various algorithms to solve the optimization problem. Simulation scenario and extensive results are presented in Section V. Finally, conclusions are drawn in Section VI.

II. Pipeline Forwarding

A. *Operating Principles*

The *pipeline forwarding* is a well-known optimal method that is widely used in computing and manufacturing. In its networking implementation, see [1] for a tutorial, all packet switches are synchronized with a *common time reference* (CTR), while utilizing a basic time period called *time frame* (TF). In a possible design UTC (coordinated universal time) can be used to derive the TF duration (T_f) from a time-distribution system such as the Global Positioning System (GPS). TFs are grouped into *time cycles* and time cycles are further grouped into *super cycles*, each super cycle lasting for one UTC second. The structure of the common time reference is depicted in Fig. 1.

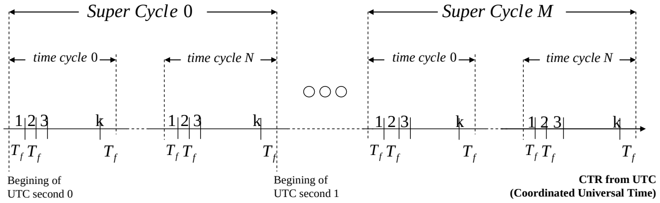

Fig. 1 The Common Time Reference structure

TFs are partially or totally assigned to each flow during a resource reservation phase. This results in a periodic schedule, repeated every time cycle, for IP packets to be switched and forwarded. The basic pipeline forwarding operation is regulated by two simple rules: (*i*) all packets that must be sent in TF t by a switch must be in its output ports' buffers at the end of TF $t-1$, and (*ii*) a packet p transmitted in TF t by switch n must be transmitted in TF $t+\tau$ by switch $n+1$, where τ is an integer constant called *forwarding delay;* TF t and $t+\tau$ are referred to as the *forwarding TF* of packet p at switch n and $n+1$, respectively. The value of the forwarding delay is determined at *resource-reservation time* and must be large enough to satisfy rule (*i*). In pipeline forwarding, a *synchronous virtual pipe* (SVP) is a predefined schedule for forwarding a pre-allocated amount of bytes during one or more TFs along a path of subsequent UTC-based switches.

[1] See for example: “Will Internet TV Crash the Internet?” on line at http://www.itnews.com.au/News/59342,web-tv-sparks-bandwidth-crisis-fears.aspx or the presentation at the OFC/NFOEC 2006 Plenary Session by Hank Kafka, Vice President for Architecture at Bell South, on the costs service providers possibly incur due to widespread deployment of video applications, on line at http://www.ofcnfoec.org/materials/2006KafkaPlenary.pdf

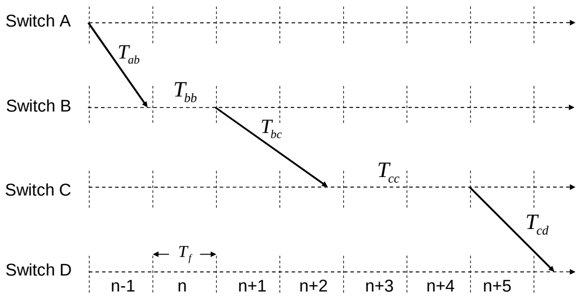

Fig. 2 Pipeline Forwarding operating principle. TF are preallocated in switches from A to D to form a synchronous virtual pipe with deterministic end-to-end delay.

As exemplified in Fig. 2, which depicts the journey of an IP packet from node A to node D along two UTC-based switches, the forwarding delay may have different values for different nodes, due to different propagation delays on different links (e.g., T_{ab}, T_{bc} and T_{cd}), and different packet processing times in heterogeneous nodes (e.g., T_{bb} and T_{cc}). Moreover, two variants of the basic pipeline forwarding operation are possible. When node n deploys *immediate forwarding*, the forwarding delay has the same value for all the packets transmitted by switch n. When implementing *non-immediate forwarding*, node n may use different forwarding delays for packets belonging to different flows. In any case, packets traveling through the network on an SVP receive a deterministic service: no packet will be lost or delayed due to congestion and the time of exit from the SVP is uniquely determined by the reserved TF in which the SVP has been entered with an uncertainty of 1 TF. Point-to-multipoint SVPs can be used to support multicast and broadcast packet delivery with guaranteed quality.

Non-pipelined (i.e., non-scheduled) IP packets, i.e., packets that are not part of a SVP (e.g., IP best-effort packets), can be transmitted during any unused portion of a TF, whether it is not reserved or it is reserved but currently unused. Consequently, links can be fully utilized even if flows with reserved resources generate fewer packets than expected. A large part of Internet traffic today is generated by TCP-based elastic applications (e.g., file transfer, e-mail, WWW) that do not require a guaranteed service in term of end-to-end delay and jitter. Such traffic can be dealt with as non-pipelined and can benefit from statistical multiplexing. Each PF node performs statistical multiplexing of best-effort traffic, i.e., inserts best-effort packets in unused TF portions. Therefore, SVPs are not at all TDM-like circuits: SVPs are virtual channels providing guaranteed service in terms of bandwidth, delay, and delay jitter, but fractions of the link capacity not used by SVP traffic can be fully utilized. Moreover, any service discipline can be applied to packets being transmitted in unused TF portions.

In summary, pipeline forwarding is a best-of-breed technology combining the advantages of circuit switching (i.e., predictable service and guaranteed QoS) and packet switching (statistical multiplexing with full link utilization) that enables a true integrated services network providing optimal support to both multimedia and elastic applications.

B. Video Transmission

Transmission of a video flow can be performed by allocating an SVP and matching the periodicity of the video frames with the periodicity of the reservation, as shown in Fig. 3. For example, if a video sequence is sampled at 30 frames per second, a super cycle lasts 1 second and contains 300 time cycles, a reservation can be made in a number of contiguous TFs {*t*, *t*+1, …, *t*+*r*} each 10 time cycles. The reservation should be large enough to enable the transmission of an encoded video frame. If a video encoder is PF-aware, the capture of video frames can be timed with the reservation so that encoding is completed just before the TF in which a reservation was made and the produced bits can be transmitted right away as a burst, which minimizes the end-to-end delay [1].

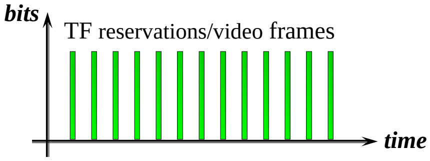

Fig. 3 Periodic bursty transmission of a video stream

However, a video stream is inherently variable as the amount of bits required to encode each video frame changes significantly. Some video encoding schemes, like MPEG, encode frames (e.g., I-frames and P-frames) with significantly different amounts of bits in a *periodic* fashion by using different techniques to eliminate spatial and temporal redundancy. Such periodic variability can be accomodated though *complex periodicity scheduling*, i.e., by allocating different amount of bits in different TFs, as shown in Fig. 4.

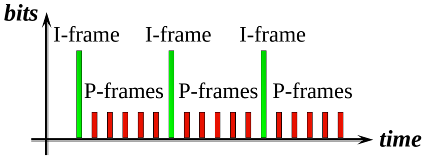

Fig. 4 Complex periodicity scheduling and reservation for transmission of a variable bit rate MPEG video stream

A significant variability still exists among video frames encoded with the same technique, e.g., among P-frames, due to the different content of the frames. [1] proposed and demonstrated a PF-aware video codec that controls the encoding process to ensure that the amount of bits produced does not exceed the reservation. This approach has two limitations: (*i*) the resulting video stream has variable quality and (*ii*) it cannot be applied with PF-unaware video encoders. This paper proposes and analyses an alternative solution where the encoder produces a variable amount of bits per video frame (possibly keeping the perceptual quality constant) and the bits exceeding the reservation in the corresponding TF are transmitted as non-pipelined traffic. Analysis-by-synthesis distortion estimation is used to select the video samples to be transmitted over the allocated SVP — i.e., with guaranteed quality of service — in such a way that user perceived quality is maximized in case the remaining samples transmitted as non-pipelined traffic are lost or delayed excessively. The presented approach overcomes the limitation of the solution presented in [1] since it (*i*) has the potential of providing more uniform perceptual quality and (*ii*) is independent of the video encoder, i.e., it can be deployed in the incremental PF introduction phase in which PF is not being used end-to-end.

III. Analysis-by-synthesis Distortion Estimation

The quality of multimedia communications over packet networks is affected by packet losses. The amount of quality degradation strongly differs depending on the perceptual importance of the lost data. In order to design efficient loss protection mechanisms, a reliable importance estimation method for multimedia data is needed. Such importance may be defined *a priori*, based on the average importance of the elements as with the data partitioning [22] approach, i.e. motion vectors are more important than residual coefficients. In order to provide a quantitative importance estimation method at a finer level of granularity, we define the importance of a video coding element, such as a macroblock or a slice (i.e. a group of macroblocks), as a value proportional to the distortion that would be introduced at the decoder by the loss of that specific element. The potential distortion of each element could be computed using the analysis-by-synthesis technique, presented in [7]. Thus a practical method to compute the distortion caused by the loss of each packet, referred to as the *distortion of the packet* in the following, is composed of the following steps:

1. Decoding, including concealment, of the bit stream simulating the loss of the packet being analyzed (synthesis stage).
2. Quality evaluation, that is, computation of the distortion caused by the loss of the packet; the original and the reconstructed picture after concealment are compared using, e.g., Mean Squared Error (MSE).
3. Storage of the distortion value as an indication of the perceptual importance of the analyzed video packet.

The previous operations can be implemented by small modifications of the standard encoding process. The encoder, in fact, usually reconstructs the coded pictures simulating the decoder operations, since this is needed for motion-compensated prediction. If the first step of the analysis-by-synthesis algorithm exploits the operations of the encoding software, complexity is only due to the simulation of the concealment algorithm. In case of simple temporal concealment techniques, this is trivial and the task is reduced to provide the data to the quality evaluation algorithm.

The analysis-by-synthesis technique, as a principle, can be applied to any video coding standard. In fact, it is based on repeating the same steps that a standard decoder would perform, including error concealment. Obviously, the importance values computed with the analysis-by-synthesis algorithm are dependent on a particular encoding, i.e. if the video sequence is compressed with a different encoder, values will be different.

Due to the inter-dependencies usually existing between data units, the simulation of the loss of an isolated data unit is not completely realistic, particularly for high packet loss rates. Every possible combination of events should ideally be considered, weighted by its probability, and its distortion computed by the analysis-by-synthesis technique, obtaining the expected distortion value. For simplicity, however, we assume that all preceding data units have been correctly received and decoded. Nevertheless, this leads to a useful approximation as demonstrated by some applications of the analysis-by-synthesis approach to MPEG coded video [9][20][21]. The effectiveness of the application of the analysis-by-synthesis technique to the packet scheduling algorithms proposed in this paper that rely on distortion values computed as described above is demonstrated by the results provided in Section V.

In this work we employ a low-complexity model-based approach, first presented in [7], to estimate the distortion caused by packet losses in future frames due to error propagation. However, note that such distortion value can also be precomputed and stored in the case of pre-recorded video (e.g. non-live streaming scenarios).

IV. Problem Formulation and Solution

A. Network Model

Fully benefiting from PF requires providing network nodes and end-systems with a Common Time Reference (CTR) and implementing PF-aware applications to maximize the quality of the received service [1]. Since this is not realistic in the near future, this work assumes PF-unaware video sources and receivers connected to portions of the network performing traditional packet switching. The generated packet stream must be time-shaped at the input of the packet forwarding network, i.e., packets are forwarded during the TFs in which resources have been allocated to their SVP. The network element implementing such functionality is called *SVP interface* (SVPI).

Each video frame is assumed to be encoded, packetized, and immediately sent by the source. For the sake of simplicity, video packets are assumed to reach the SVPI without loss and after a negligible delay. This model is realistic in the currently common scenario of a lightly loaded (asynchronous) broadband access network. Consequently, all packets belonging to a video frame are assumed to be available at the SVPI every $1/f_r$ seconds, where f_r is the frame rate of the video sequence. (See Table I for a summary of the notation adopted throughout the paper.)

In this paper we refer to loss/late probability P_{ll} as the probability of a packet either being lost or not arriving at the receiver on time for playback. Based on the above assumptions, we model an SVP, as an *independent time-invariant channel with deterministic constant delay and loss/late probability equal to zero*, as proposed in [10]:

$$P_{ll}^{PF} = 0\,. \tag{1}$$

. The end-to-end delay on the SVP τ_{PF} can be calculated as follows[2]:

$$\tau_{PF} = \sum_{i=1}^{N}\left(\left\lceil \frac{Cd_i}{T_f}\right\rceil \cdot T_f + T_f\right) + Cd_{N+1} + J \tag{2}$$

where N is the number of PF nodes on the path, Cd_i is the propagation delay between the $(i-1)_{th}$ node and the i_{th} node (the ingress SVPI being node 0 and the egress SVPI being node N+1, T_f is the duration of the TF and J is the jitter, $0 \le J \le T_f$ (see [1] for further information on jitter characterization). Given that in normal operating conditions T_f is at most on the order of hundreds of microseconds, for all purposes of video transmission the end-to-end delay can be considered constant and deterministically upper-bounded, given the path the video flow takes through the network, as:

$$\tau_{PF} = \left(\sum_{i=1}^{N}\left\lceil \frac{Cd_i}{T_f}\right\rceil + N + 1\right)\cdot T_f + Cd_{N+1}\,. \tag{3}$$

The non-pipelined channel is modeled as an *independent time-invariant packet drop channel with random delay*: a non-pipelined packet sent at time t experiences: (*i*) a loss probability p_{lost}^{NPF} independent of *t*, and (*ii*) a variable end-to-end delay τ_{NPF} whose distribution is $f_\tau(t \mid not\ lost)$. The same model has been proposed in [11].

In order to estimate the loss/late probability of non-pipelined channel p_{ll}^{NPF}, loss probability and end-to-end delay distributions must be known either by devising a mathematical model (which will be object of further research) or by inferring internal network behavior from external end-to-end network measurements. The latter, often referred to as network tomography [13],[14],[15], is based on processing of packet traces and accurate timestamps associated to packets. Availability of a CTR to devise coherent timestamps can be beneficial to the network tomography process. The following analysis is independent of the method used to estimate the statistical behavior of the non-pipelined channel.

Let t_0 and t_{dd} be the forwarding time of a packet and its *delivery deadline* at the receiver, respectively. We define $\theta = (t_{dd} - t_0)$ as the maximum time packet can spend in the network.

TABLE I

SYMBOLS

f_r	Video frame rate.
T_f	Time frame size.
Cd_i	Propagation delay between the $(i-1)_{th}$ node and the i_{th} node.
τ_{PF}	End-to-end delay of the whole PF channel.
τ_{NPF}	End-to-end delay of the non pipelined channel.
P_{ll}^{PF}	Loss late probability on the PF channel.
P_{ll}^{NPF}	Loss late probability on the non-pipelined channel.
$P_{ll}^{t_i}$	Loss late probability on the non-pipelined channel given that packets is forwarded at time t_i.
$t_{fd,i}$	Forwarding deadline of packets belonging to frame i
$t_{dd,i}$	Delivery deadline of packets belonging to frame i
θ	Maximum time a packet can spend in the network.
τ_A	Maximum end-to-end delay tolerated by the application.
τ_s	Maximum delay the SVPI can insert while satisfying end-to-end application requirements.
d_0	Distortion caused by the encoding process.
d_i	Packet distortion in case of loss.
Ω	The whole set of packets in which the video sequence is packetized.
$\alpha_{VFi}^{'}$	Subset of packets, belonging to video frame *i*, scheduled for pipeline forwarding at the first scheduling optimization.
$\alpha_{VFi}^{''}$	Subset of packets, belonging to video frame *i*, scheduled for pipeline forwarding at the second scheduling optimization opportunity.
$\beta_{VFi}^{'}$	Subset of packets, belonging to video frame *i*, forwarded as non-pipelined after the first scheduling optimization.
$\beta_{VFi}^{''}$	Subset of packets, belonging to video frame *i*, forwarded as non-pipelined after the second scheduling optimization.
ω_{VFi}	Subset of least perceptually important packets of the video frame *i*, delayed for further scheduling optimizations at the SVPI.

Thus we formulate the loss/late probability as follows:

$$p_{ll}^{NPF} = P\{\tau_{NPF} > \theta\}\ ,$$

$$p_{ll}^{NPF} = p_{lost}^{NPF} + (1 - p_{lost}^{NPF})\int_{\theta}^{\infty} f_\tau(t \mid not\ lost)\, dt\,. \tag{4}$$

B. Expected Distortion

The video encoding process introduces a distortion d_0 that depends only on the encoding algorithm. However, when video sequences are transmitted, errors and packet losses contribute to increase such distortion. The exact expected distortion value at the receiver for a given video sequence should be computed as the weighted average of the distortions

[2] Without loss of generality, the presented end-to-end delay calculation assumes the time required by network nodes to move packets from input to output to be zero. See [1] for details on the pipeline forwarding end-to-end delay calculation.

corresponding to all the possible realizations of the network channel, the weight being the probability of a specific channel realization, as formulated in [11]. However, this procedure being impractical due to its computational complexity, a linear approximation is commonly used [11][23][6][21]:

$$E[d] = d_0 + \sum_{i=1}^{N} d_i \cdot p_l^i \, .$$

where d_i is the distortion that the loss of the i^{th} packet would introduce, p_l^i is the probability of losing that packet and N the total number of packets in which the video sequence is transmitted. In other words, it is assumed that if two packets p_1 and p_2 have distortion d_1 and d_2, respectively, their loss causes an overall distortion d_1+d_2.

Let us define Ω as the set of packets in which the video sequence is packetized, α the subset of packets transmitted on an SVP and β the subset of non-pipelined packets such that $\Omega = \alpha \cup \beta$ and $\alpha \cap \beta = 0$. Thus the expected distortion can be written as:

$$E[d] = d_0 + \sum_{i\in\alpha} d_i \cdot p_{ll}^{PF} + \sum_{j\in\beta} d_j \cdot p_{ll}^{NPF} \, .$$

Being the loss/late probability of an SVP zero, we obtain:

$$E[d] = d_0 + \sum_{j\in\beta} d_j \cdot p_{ll}^{NPF} \, . \qquad (5)$$

Two independent optimization steps can be implemented at the SVPI to minimize $E[d]$ at the receiver: (*i*) using PF for the transfer of packets with highest d_i and (*ii*) minimizing the loss/late probability p_{ll}^{NPF} experienced by non-pipelined packets.

C. Selection of packets for transfer through an SVP

As mentioned before, the SVPI can compute in advance the delay experienced by packets transferred on an SVP. This, the SVPI assigns a *forwarding deadline* to each packet, which is the latest time at which the packet can be forwarded to arrive on time for playback at the receiver.

Also, transmission opportunities for packets are known in advance given the set of TFs in which a reservation for the corresponding video flow exists and the amount of bits that can be transmitted in each allocated TFs. Thus the SVPI schedules for transmission on an SVP the most perceptually important packets (i.e. packets with the highest d_i) that have a transmission opportunity before their forwarding deadline. The remaining packets (i.e., the least perceptually important packets) are transmitted by the SVPI as non-pipelined traffic. The advance knowledge of the forwarding deadlines and transmission opportunities enable the SVPI to perform scheduling optimizations as soon as packets arrive at the SVPI so that packets that do not have a chance to be transmitted on the corresponding SVP by their forwarding deadline can be immediately forwarded as non-pipelined traffic. This increases the probability of their timely arrival to the receiver because they can spend more time traversing the network, thus minimizing the expected distortion.

1) Problem formulation

Given the arrival time t_i of a video frame at the SVPI, the maximum end-to-end delay tolerable by the application τ_A, and the maximum latency through the SVP τ_{PF} , the maximum value of the forwarding deadline $t_{fd,i}$ for each packet belonging to video frame *i* is calculated as follows

$$t_{fd,i} = t_i + (\tau_A - \tau_{PF}) \, . \qquad (6)$$

$\tau_s = \tau_A - \tau_{PF}$ is the maximum time pipelined packets can spend at the SVPI while still satisfying (6). τ_s also is the maximum time packets sent through an SVP spent at the receiver before their video frame is played, i.e., τ_s together with the bit rate allocated for the SVP provide the basis for the calculation of the minimum size of the play-out buffer that guarantees enough storage for received pipelined packets until their play-out deadline.

We formulate the problem of determining which are the best packets α_{VFi} to transmit in each allocated TF as a zero-one multiple knapsack problem [16],[17],[18]. Given n packets of the video frame, each having distortion d_j and size s_j , and m allocated TFs before the forwarding deadline for the i^{th} video frame, c_i bits reserved in each, the zero-one multiple knapsack problem consists in assigning packets to the *m* allocated TFs in such a way that the total distortion of the assigned packets is maximized, the total amount of packets assigned to each TF does not exceed the allocation, and each packet is either assigned to one of the TF or treated as non-pipelined packet. More formally:

$$\begin{aligned} &\textit{Maximize} \sum_{i=1}^{m}\sum_{j=1}^{n} d_i x_{ij} \\ &\textit{Subject to} \sum_{j=1}^{n} s_j x_{ij} \le c_i \quad i = 1,\ldots,m, \\ &\qquad\qquad \sum_{i=1}^{m} x_{ij} \le 1 \qquad j = 1,\ldots,n, \\ &\qquad\qquad x_{ij} \in \{0,1\} \qquad i = 1,\ldots m, \;\; j = 1,\ldots,n \;, \end{aligned} \qquad (7)$$

where $x_{ij} = 1$, if packet *j* is assigned to TF i, and $x_{ij} = 0$, otherwise. Moreover, s_i, d_i and c_i are required to be positive integers and for each packet there must be at least one TF with enough allocated capacity during which the packet can be transmitted, that is the maximum amount of reserved bits c_i across all allocated TFs must be greater than the maximum packet size and the minimum amount must be larger than the minimum size packet. Formally:

$$\max_{j=1,\ldots,n} s_j \le \max_{i=1,\ldots,m} c_i \tag{8}$$

$$\min_{i=1,\ldots,m} c_i \ge \min_{j=1,\ldots,n} s_j \; . \tag{9}$$

This problem is known to be NP-hard, that is, the time required to find the exact solution can be exponential in the input size. Several heuristic algorithms have been developed to find sub-optimal solutions with reduced complexity, especially in case of particular problem structures. In this work we adopted two simple heuristic algorithms, whose main advantage is their low complexity, and compared their performance with the optimal solution computed by exhaustive search. The algorithms are described in Section IV.C.3).

2) *General solution: the Distortion Optimized Scheduling Algorithm (DOSA)*

This paper generally refers to the algorithm to find the best solution to the above optimization problem as a *distortion optimized scheduling algorithm* (DOSA). Finding a global optimum minimizing the expected distortion requires the DOSA to consider the entire video sequence, which is computationally demanding and also not feasible in a live video (real-time) scenario. In order to avoid packets to arrive at the receiver beyond their delivery deadline because of the delay introduced by both the network and SVPI, the DOSA is run on a small part of a video sequence, which results in a locally optimal schedule. The length of the video sequence on which the algorithm is run is determined based on the latency through the allocated SVP τ_{PF} and the maximum end-to-end delay τ_A allowed by the application, or a percentile thereof.

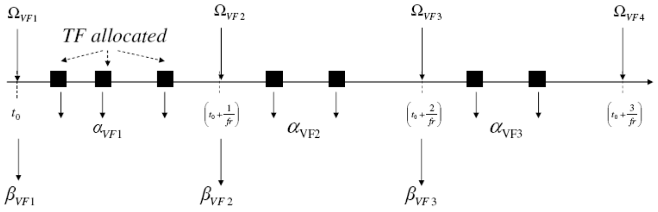

Fig. 5 Example of pipelined packet selection optimization step by the DOSA run over one video frame period. Ω_{VFi} is the set of packets carrying the i^{th} video frame , α_{VFi} is the set of packets scheduled for pipeline forwarding and β_{VFi} is the set of packets sent as non-pipelined traffic.

Fig. 5 shows a sample scenario in which the SVPI sets the forwarding deadline of a packet belonging to frame *i* to the time of arrival of frame *i* at the SVPI plus the video frame period[3], i.e. $t_{fd,i} = t_i + (1/f_r)$, assuming that this value is compatible with the application delay requirements.

In the scenario depicted in Fig. 5, at time t_0 the SVPI computes (*i*) the forwarding deadline $t_{fd,1}$ for each packet of the first video frame and (*ii*) the total amount of bits allocated in the TFs before $t_{fd,1}$. The proposed approach is to schedule for PF the subset of packets α'_{VF1}, that maximizes the utilization of the TF reservation from the distortion viewpoint. After the scheduling optimization the SVPI forwards as non-pipelined traffic the subset β'_{VF1} of least perceptually important packets immediately — i.e., without waiting for the subset α'_{VF1} of packets to be sent out. Those packets are handled on a FIFO basis. Note however that any other forwarding policy (e.g., burstiness, prioritization, etc.) can be applied to non-pipelined traffic without affecting the validity of the analysis presented in this work.

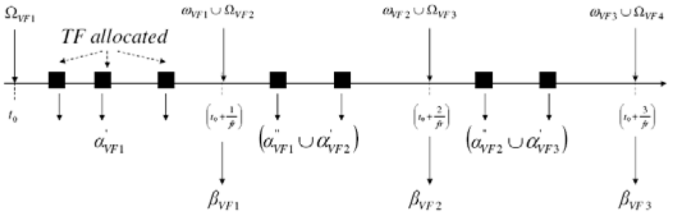
Ω_{VF1}
$\omega_{VF1} \cup \Omega_{VF2}$
$\omega_{VF2} \cup \Omega_{VF3}$
$\omega_{VF3} \cup \Omega_{VF4}$
TF allocated
t_0
α'_{VF1}
$\left(t_0+\frac{1}{fr}\right)$
$\left(\alpha''_{VF1} \cup \alpha'_{VF2}\right)$
$\left(t_0+\frac{2}{fr}\right)$
$\left(\alpha''_{VF2} \cup \alpha'_{VF3}\right)$
$\left(t_0+\frac{3}{fr}\right)$
β_{VF1}
β_{VF2}
β_{VF3}

Fig. 6 Example of pipelined packet selection optimization step by the DOSA run over two video frame periods.

Fig. 6 shows a sample scenario in which the SVPI sets the forwarding deadline of a packet belonging to frame *i* to the time of arrival of the frame at the SVPI plus twice the video frame period, i.e. $t_{fd,i} = t_i + (2/f_r)$, assuming that this value is compatible with the application delay requirements. In this scenario DOSA optimizes the transmission schedule over two video frames. As depicted in Fig. 6 at time t_0 the DOSA schedules for forwarding through the SVP the most perceptually important packets α'_{VF1} of the first video frame during the TFs allocated before $t_0 + (1/f_r)$. Note that the prime and double prime are used to identify packets, carrying video samples of the same frame, scheduled for PF at the first and at the second scheduling times, respectively. At time $t_0 + (1/f_r)$ DOSA runs on the remaining packets of the first video frame $\omega_{VF1} = \Omega_{VF1} \setminus \alpha'_{VF1}$ and all packets of the second video frame Ω_{VF2}. The subset $\alpha''_{VF1} \cup \alpha'_{VF2}$ of the most perceptually important packets belonging to either the first or the second video frame are scheduled for transmission on the SVP.

[3] Note that in general the forwarding deadline for the packets carrying a given video frame does not need to be aligned with the arrival time of the next video frame.

The remaining subset of packets $\beta_{VF1} = \Omega_{VF1} \setminus \alpha'_{VF1} \setminus \alpha''_{VF1}$ belonging to the first video frame non scheduled for PF in the last useful TFs, i.e., the TFs allocated before the forwarding deadline $t_{fd,i} = t_i + (2/f_r)$, are forwarded as non-pipelined traffic. The subset ω_{VF2} of least perceptually important packets of the second video frame are reconsidered in the selection at time $t_0 + (2/f_r)$, when also packets of the third video frame Ω_{VF3} become available.

3) *DOSA flavors*

Based on the previous analysis, three scheduling algorithms have been implemented to determine the best packets to transmit at each transmission opportunity.

First, we implemented an exhaustive searching algorithm over all possible schedules, determining the one which minimizes the expected distortion, thus solving (7). To reduce the implementation complexity a branch and bound approach has been adopted during the search, to discard invalid solutions as soon as possible. This algorithm is referred to as the *Minimum Distortion Algorithm* (MDA).

Moreover, we also used two low-complexity heuristic algorithms and we compared their performance as shown in Section V. Packets to be forwarded through an SVP are considered in decreasing order of distortion (D_i). Given the number of TFs allocated and the allocation size during each TF, the first algorithm schedules each packet during the first available TF with enough reserved capacity. This algorithm is referred to as the *First Fit Algorithm* (FFA).

The second heuristic algorithm assigns each packet to the TF with the largest amount of free space. This will be referred to as the *Worst Fit Algorithm* (WFA).

Note that packet reordering issues stemming from the deployment of the above algorithms can be handled at the receiver by means of the RTP sequence number.

D. *Minimizing loss probability of non-pipelined packets*

If on the one hand running the DOSA on a large part of a video sequence might increase the amount of packets sent through the corresponding SVP, on the other hand it delays the forwarding of packets that are finally selected for non-pipelined service, thus lowering their possibility of making it to the receiver by the play-out deadline of their video frame. In order to reduce the expected distortion, a second optimization step can be implemented at the SVPI to delay only packets with high probability of being selected for forwarding through the SVP at the next scheduling opportunities. The improved algorithm is referred to as *delay and distortion optimized scheduling algorithm* (DDOSA).

Fig. 7 shows a sample scenario in which the SVPI sets the forwarding deadline of a packet belonging to frame *i* to the time of arrival at the SVPI plus twice the video frame period, i.e. $t_{fd,i} = t_i + (2/f_r)$, assuming that this value is compatible with the application delay requirements. At time t_0 packets of the first video frame are available at the SVPI and the DDOSA schedules for transmission the subset of packets α'_{VF1}, that maximizes the PF channel utilization from the distortion viewpoint, as with the DOSA algorithm. Among the set of remaining packets $\Omega_{VF1} \setminus \alpha'_{VF1}$, DDOSA determines the subset of packets β'_{VF1} with the least probability of being selected for transmission through the SVP at the next scheduling optimization opportunities; these packets are transmitted as non-pipelined traffic immediately. The remaining subset of packets $\omega_{VF1} = \Omega_{VF1} \setminus \alpha'_{VF1} \setminus \beta'_{VF1}$ are delayed, to be considered at the next scheduling optimization opportunity.

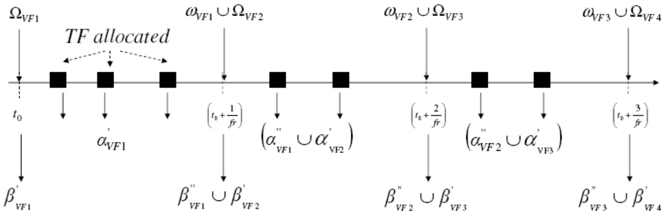
Ω_{VF1}
TF allocated
$\omega_{VF1} \cup \Omega_{VF2}$
$\omega_{VF2} \cup \Omega_{VF3}$
$\omega_{VF3} \cup \Omega_{VF4}$
t_0
α'_{VF1}
$\left(t_0 + \frac{1}{fr}\right)$
$\left(\alpha''_{VF1} \cup \alpha'_{VF2}\right)$
$\left(t_0 + \frac{2}{fr}\right)$
$\left(\alpha''_{VF2} \cup \alpha'_{VF3}\right)$
$\left(t_0 + \frac{3}{fr}\right)$
β'_{VF1}
$\beta''_{VF1} \cup \beta'_{VF2}$
$\beta''_{VF2} \cup \beta'_{VF3}$
$\beta''_{VF3} \cup \beta'_{VF4}$

Fig. 7 DDOSA Example.

Hence an algorithm is needed to estimate the probability that a packet is being pipeline forwarded at the next scheduling opportunity, depending on the packet distortion d_i. Obviously, since the selection of packets for PF by the DDOSA at future optimization opportunities depends on the distortion value of packets not yet available at the SVPI when the DDOSA makes its choice, part of the ω_{VF1} packets, namely β''_{VF1}, might be anyway forwarded as non-pipelined traffic at the next scheduling opportunity.

However, to evaluate the improvements of the DDOSA scheme independently of the probability estimation method, an omniscient algorithm has been implemented and tested in this work. This algorithm has perfect knowledge of the number, size and distortion of future packets arriving at the SVPI. Thus, it can exactly predict which packets in subset $\Omega_{VFi} \setminus \alpha'_{VFi}$ will be pipeline forwarded at the next transmission opportunities and consequently it immediately forwards the remaining packets as non-pipelined traffic without further delay.

The performance of the omniscient algorithm represents the upper bound of the performance achievable with this transmission scheme. Since simulations show that performance improvements are higher than 2 dB PSNR (see Section V.B), we are currently working on a heuristic algorithm, based on a model of packet size and distortion of future frames, which estimates the probability that a certain packet in subset $\Omega_{VFi} \setminus \alpha'_{VFi}$ will be forwarded over an SVP at the next transmission opportunities. Depending on this probability, the algorithm decides to delay packets or forward them immediately as non-pipelined traffic.

V. Simulation Results

A. Simulation Setup

The proposed algorithms have been implemented and simulated with the well-known network simulator *ns-2* on a typical bottleneck topology. Bottleneck bandwidth is set to 10 Mbit/s, with a 50 ms propagation delay. The time required for the transfer from source nodes to the SVPI is assumed to be 5 ms, as well as from the egress of the SVP to the destination nodes. The bandwidth of those links is oversized not to impact on the results, which is reasonable in the currently realistic scenario of a lightly loaded (asynchronous) broadband access network. The video transmissions assume the use of the IP/UDP/RTP protocol stack, which is commonly deployed for real-time multimedia communications. Interfering exponential cross traffic was also considered. Depending on the simulations, its rate has been varied from 2 to 4 Mbit/s.

An H.264 [19] software encoder was used with three different video encoding schemes that achieve an increasingly smoother bitrate profile at the expense of video quality. The first scheme, called *standard scheme* (Std.) in this work, encodes the video sequence as a repeated pattern of one I-type frame followed by eleven P-type frames. The quantization step-size is fixed, hence video quality is approximately constant. This scheme presents high bandwidth peaks in correspondence of I-type frames, but provides the highest encoding quality. The second scheme, called *intra-refreshed (IR)*, uses all P-type frames. However, to reduce error propagation in case of packet losses an intra refresh method is employed, which refreshes the whole picture one slice at a time (i.e., one slice is encoded as in an I-type frame: not by reference to a previous frame), achieving a full picture refresh with the same frequency of I-type frames in the first scheme. In this case, too, the quantization step-size is fixed. The average bit-rate is similar to the previous case, however the rate profile is smoother since I-type frames are absent but encoding quality is lower. The last encoding scheme, called *intra refresh and rate control (IR+RC)*, uses the same intra refresh approach of the previous one, but it also employs the rate control algorithm included in the standard H.264 software, set to achieve approximately the same average bit-rate of the previous two schemes. The quality is worse than the one achieved by the previous schemes. Given an average bitrate, in fact, the standard scheme has maximum freedom in allocating bits for video frames, hence frames which are particularly complex are given much more bits than others. The IR+RC scheme, instead, having a fixed amount of bits for each frame, is compelled to reduce the quality in case of complex frames, which in turn causes an efficiency reduction of the prediction mechanism between frames.

We used three video sequences known as *foreman, mad* and *lts* encoded at CIF resolution (352x288), 30 fps, at high quality. Video quality has been evaluated using the PSNR (peak signal-to-noise ratio) distortion measure, which, taking in due consideration its well-known limits, is widely accepted in the multimedia communications research community. The encoding quality ranges from 35.8 to 36.4 dB PSNR for the *foreman* sequence, from 35.8 to 36.7 dB for the *lts* sequence and from 37.0 to 38.0 dB for the *mad* sequence, depending on the encoding scheme. All experiments involve sending several video sequences at the same time. Results always show the mean of the PSNR computed over the three test video sequences. Moreover, results always analyze the impact of the encoding scheme on the performance (PSNR).

Fig. 8 shows the bit-rate profiles for the three video encoding schemes, for the *foreman* video sequence. The other sequences present similar characteristics. The IR+RC scheme achieves the smoothest bit-rate, while the standard scheme presents the maximum variability. The IR scheme presents an intermediate behavior; moreover, the bit-rate can significantly vary in the long term since it depends on the video content. Note that the smoothness of the bit-rate profile is in inverse relation to the encoded video quality. This is indeed a well-known trade-off in video coding.

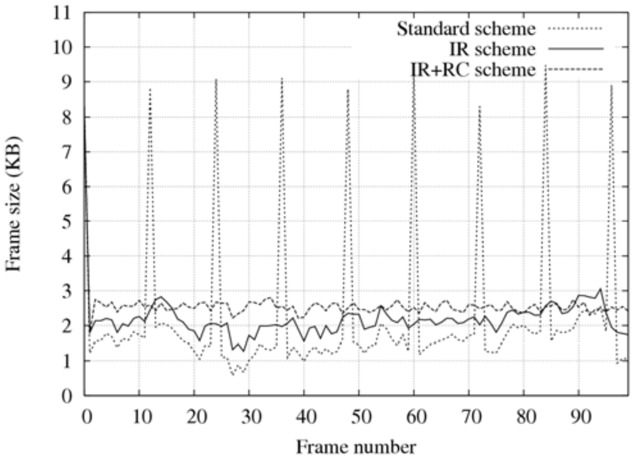

Fig. 8 Frame size as a function of frame number for the three analyzed coding schemes. Foreman sequence. Average bitrate is similar in all three cases, about 620 kbit/s.

B. Scheduling Algorithms

This section presents the performance of the proposed video transmission algorithms over a network implementing PF. The first set of video transmission simulations compare the performance of the three proposed DOSA scheduling algorithms in a scenario featuring fifteen video flows with 4 Mbit/s concurrent interfering traffic. In this scenario the offered network load is about 2 Mbit/s higher than the bottleneck link capacity and the amount of bandwidth allocated to the SVP for each video flow is equal to its average bit-rate.

Fig. 9 shows the performance of the various scheduling algorithms. As expected, the optimum scheduling algorithm (MDA) outperforms the other techniques. Depending on the video encoding scheme, the gain ranges from 1 to 2 dB PSNR. However, such performance decrease is traded for reduced algorithm complexity. The performance of the FFA and WFA algorithm is very similar, since in our experiments the PSNR difference never exceeds 0.02 dB.

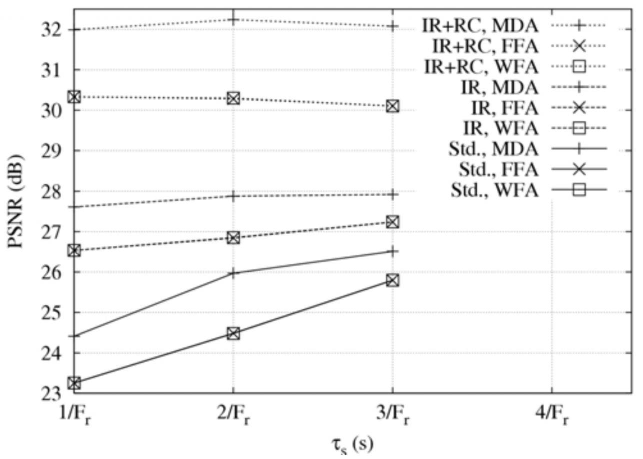

Fig. 9 PSNR of the three proposed algorithms, as a function of the sender side delay.

When τ_s, i.e. the maximum time pipelined packets can spend at the SVPI while satisfying the application maximum delay requirement, is increased, DOSA runs on a higher number of packets at the SVPI, particularly for the case of the standard scheme that presents the most irregular instantaneous bit-rate. All scheduling algorithms take great advantage of the larger time window for packet forwarding. Most of the packets belonging to I-type frames, in fact, present a high distortion value but, due to their large size, they cannot generally be sent before the next frame arrival at the SVPI, hence they are delayed waiting for the next PF transmission opportunity. A higher sender side delay allows to spread high-distortion packets on a larger time period, thus increasing the performance, up to 2 dB in case of the standard encoding scheme. For the case of the IR scheme, the gain is reduced to 1dB while there is no perceptible performance variation for the case of the IR+RC scheme.

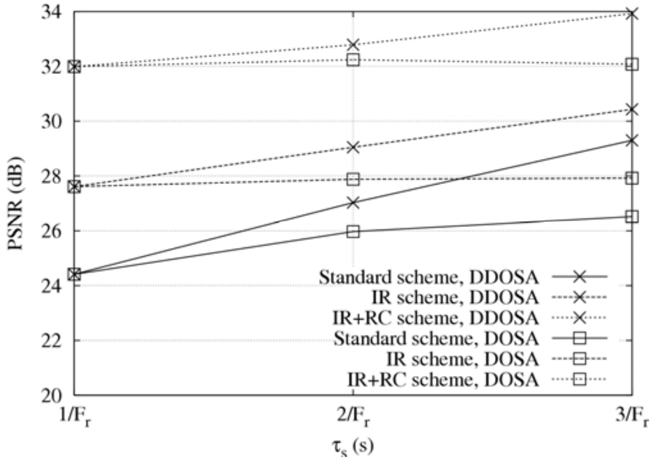

Fig. 10 PSNR comparison with DOSA and withDDOSA.

The second set of simulations shows the improvement that can be achieved by employing both optimization steps, i.e. by deploying DDOSA. In this case, not all packets that are not going to be sent on the SVP before the next video frame are kept in the SVPI waiting for the next scheduling decision. DDOSA estimates which packets present the highest probability of not being sent with PF at the next transmission opportunity and it immediately sends them as non-pipelined traffic. In the simulations an omniscient technique has been used, i.e., the algorithm knows and uses the distortion values of packets carrying future video frames not yet arrived at the SVPI. This approach allows to assess the maximum performance that can be achieved by DDOSA. Fig. 10 shows the PSNR of transmissions employing both DOSA and DDOSA, compared to performance of the DOSA only. The results show considerable improvements, ranging from 2 to 3 dB PSNR, if τ_s is increased to three video frame periods. The improvement is larger in the case of the standard encoding scheme, while it is smaller in the case of the IR+RC scheme.

C. Bandwidth Allocation Strategies

In the previous experiments, for each video sequence, the average size of the video frames was reserved during a number of TFs each video frame period. However, this might not be the optimal choice. In fact, due to the irregularity of the instantaneous bandwidth of the video sequences especially in case of periodic insertion of I-type frames, as shown in Fig. 8, the amount of packets which can be sent as PF traffic strongly varies over time. This section studies the tradeoffs involved in choosing both encoding bit-rate and SVP bandwidth allocation in a scenario in which only the DOSA algorithm is being deployed (i.e., where only a single optimization step is implemented)

Given the SVP bandwidth allocated for a certain video sequence, the problem is to find the video encoding rate that provides the best performance, considering that instantaneous bandwidth irregularities will prevent the transmission of all packets as pipeline forwarded traffic. Instantaneous bandwidth irregularities are mainly due to the presence of different types of frames — in the case of the standard encoding scheme — and variations of the characteristics of the input signal. The rate-controlled video presents the least level of irregularities, however small variations are still present even in this case.

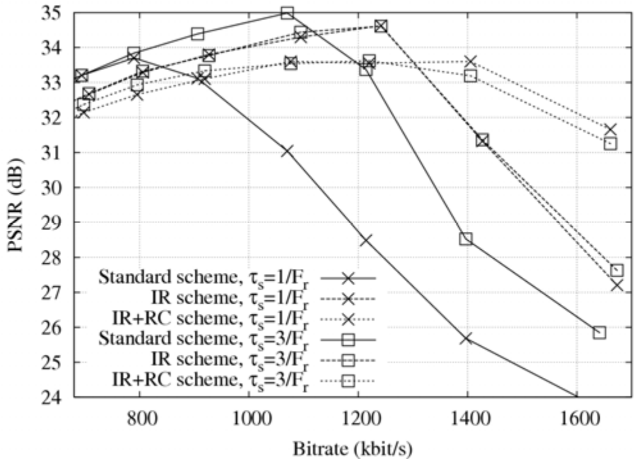

Fig. 11 PSNR performance of the MDA as a function of the aggregated bit-rate, i.e. the sum of the average bitrates of the three tested video sequences. The bandwidth allocated to each video sequence equals the average frame size.

Fig. 11 shows the PSNR performance of the MDA as a function of the aggregated bit-rate, i.e. the sum of the average bitrate of the three tested video sequences. For all these experiments the bandwidth reserved for each SVP is about 630, 730, 300 kbit/s for the *foreman*, *lts* and *mad* sequences respectively., i.e. the average bitrate of the sequences coded at the highest quality employed in the experiments. This allocation allows fifteen video flows to be transferred on the bottleneck link in our simulations. Note that while the allocated bandwidth is constant, the video encoding rate has been progressively decreased (moving in figure from right to left). Moreover, 2 Mbit/s of exponential interfering traffic was injected in the network to simulate a highly loaded network scenario.

In the left part of the figure, the PSNR performance approximates the encoding distortion, since nearly all packets can be sent as pipeline forwarding traffic. In this condition, as previously explained, the encoding quality is the highest for the standard scheme and the lowest for the IR+RC scheme.

Each curve presents a point of maximum, which corresponds to the optimal trade-off between bandwidth dedicated to video encoding and the SVP bandwidth reserved. Increasing the encoding bandwidth, in fact, improves video quality, but if a certain threshold is exceeded, the effect of packet losses on the fraction of packets which are sent as non-pipelined traffic reduces the performance. This effect appears at lower bitrates for the standard encoding scheme since video packet size is more likely to exceed the amount of allocated bandwidth due to I-type frames. For the case of IR and IR+RC schemes, the value of aggregated bandwidth that maximizes the PSNR performance is about 1,200 kbit/s, i.e. the encoding bit-rate should be about 75% of the priority-forwarding allocated bandwidth.

Note that, unlike the previous two cases, for the standard encoding scheme, the value that maximizes the PSNR performance strongly depends on τ_S. The video transmission algorithm, in fact, suffers from the presence of I-type frames, which present a much larger size that the other frames. Hence, a significant part of the I-type frames is often sent as non-pipelined traffic, which may strongly affect the video quality performance in case of packet losses. However, if the sender side delay is increased, the quality increases because the perceptually important I-frame packets can be spread over a larger number of TFs. In this condition, despite the lower encoding rate compared to the other two encoding schemes, the PSNR performance is improved in case of τ_S equal to three video frame periods.

To demonstrate the efficiency of the proposed transmission algorithms for the case of allocation based on the average frame size, Fig. 12 compares the previous results with a different allocation, i.e. we allocated an amount of bandwidth for each video flow such that 90% of the video frames are smaller than the allocation during a video frame period. This is equivalent to allocate 900 kbit/s for the *foreman* and *lts* sequences, and 300 kbit/s for the *mad* sequence. We did not allocate the peak bandwidth of the video flows because, considering the observation of the cumulative distribution function of frame sizes, depicted in Fig. 13, reserving bandwidth based on 90% of the video frames seems a good trade-off: capturing the remaining 10%, in fact, would approximately triple the needed allocation, which in turn would drastically reduce the number of video flows that can be carried in the network.

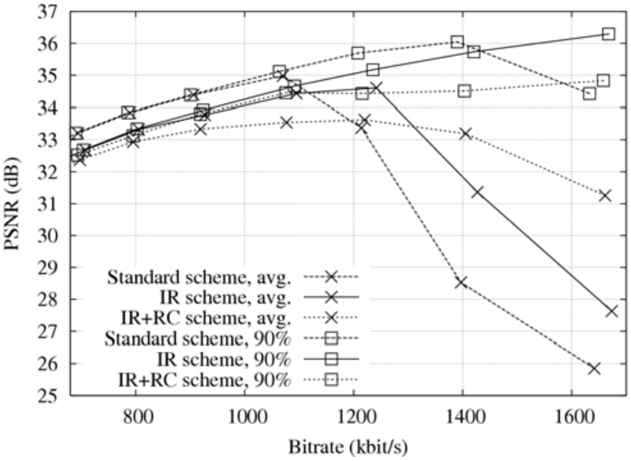

Fig. 12 PSNR performance of the MDA as a function of the aggregated bit-rate, i.e. the sum of the average bitrates of the three tested video sequences. SVP bandwidth allocation is based either on the average frame size or the 90 percentile of the frame size. τ_S set to 3/F_r after packet arrival to the SVPI.

As in the previous experiments, the absolute value of the allocated bandwidth is then kept constant while the video encoding rate has been progressively decreased (moving in figure from right to left). However, unlike the previous case, with the allocation based on the 90 percentile of video frame size a very large part of the video packets can fit into the reserved timeframes even when the encoding bit-rate is high. Note also that, to allow such larger bandwidth allocation, the number of video flows has been decreased to twelve. As expected, a larger allocation leads to better PSNR results: the performance increase between the point of maximum of the two allocations is about 1.5 dB for the IR+RC scheme, 2 dB for the IR scheme and 1 dB for the standard scheme.

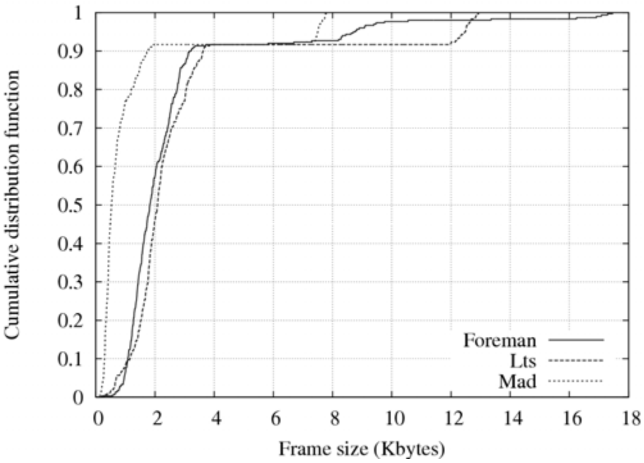

Fig. 13 Cumulative distribution function of the frame sizes, for *foreman*, *lts*, *mad* sequences, coded at about 620 kbit/s, standard encoding scheme, H.264 codec JM v.11.0.

These results also demonstrate that with the proposed scheduling algorithm, if a moderate decrease of the PSNR performance is acceptable with respect to the case of 90% allocation, the pipeline-forwarding bandwidth for each flow can be reduced so that 25% more video communications are possible in the same network conditions. In any case, even though the efficiency in the utilization of network resources is lower with the 90 percentile-based allocation — 8.4 Mb/s are reserved for 12 video flows, rather than for 15 as in the average frame size allocation policy —, it is anyway higher than the one commonly achieved on networks where quality of service is based on the deployment of differentiated services. In fact, being common practice to reserve at most 20-40% of the network capacity to delay sensitive traffic, at most 9 video flows[4] would be allowed through a DiffServ network equivalent to the bottleneck topology deployed in the simulations.

VI. Conclusions

This work presents a quality-oriented multimedia delivery framework aimed at optimizing video transmission for both interactive and broadcasting applications over packet networks with respect to both resource utilization and user perceived quality. The work explores the issue of multimedia packet scheduling for delivery using pipeline forwarding. Two heuristic scheduling algorithms based on the use of the perceptual information of the carried video samples are proposed. Performance bounds have been established by comparing the proposed algorithms with an exhaustive-search approach, showing that performance is within 2 dB PSNR from the optimal solution in the worst case. Also, the impact of applying a double-optimization-step scheduling algorithm versus a single-optimization-step one have been simulated quantifying the advantage of the former. The work also investigated the impact of encoding schemes with different tradeoffs between encoded video quality and bitrate fluctuations, showing their impact on performance. Finally, bandwidth allocation issues have been experimentally studied, showing the tradeoffs between quality and reserved bandwidth, i.e., ultimately the maximum number of video flows that can be carried by the network. Future work will analytically study allocation tradeoffs in case of multiple flows, as well as the optimal multimedia encoding rate given a pipeline forwarding allocation. Moreover, a double-optimization-step scheduling algorithm that does not require advanced knowledge of the whole encoded video sequence is being developed for real-time operation.

[4] Since the three different video sequences have an average aggregated bandwidth of 1660 Kb/s, 9 video flows have an average aggregate bandwidth of 4980 Kb/s, or about 50% of the bottleneck capacity, which exceeds the fraction of link capacity usually dedicated to delay sensitive traffic.

References.

[1] M. Baldi, Y. Ofek, “End-to-end Delay of Video-conferencing over Packet Switched Networks,” *IEEE/ACM Transaction on Networking*, vol.8, no.4, pp. 479-492, Aug. 2000.

[2] C-S. Li, Y. Ofek, A. Segall, K. Sohraby, “Pseudo-Isochronous Cell Forwarding,“ *Computer Networks and ISDN Systems*, vol.30, no.24, pp. 2359-2372, Dec. 1998.

[3] M. Baldi, Y. Ofek, "Blocking Probability with Time-driven Priority Scheduling," *SCS Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2000)*, Vancouver, BC, Canada, July 2000.

[4] M. Baldi, Y. Ofek, B. Yener "Adaptive Group Multicast with Time-Driven Priority," *IEEE/ACM Transactions on Networking*, Vol. 8, No.1, Feb. 2000, pp. 31-43.

[5] M. Baldi, Y. Ofek, "Multi-Terabit/s IP Switching with Guaranteed Service for Streaming Traffic," *IEEE High-Speed Networking Workshop at INFOCOM 2006*, Barcelona (Spain), Apr. 2006.

[6] E. Masala, J.C. De Martin, “Analysis-By-Synthesis Distortion Computation For Rate-Distortion Optimized Multimedia Streaming”, *IEEE Int. Conf. on Multimedia & Expo (ICME)*, Baltimore, MD, vol. 3, pp. 345-348, Jul. 2003.

[7] F. De Vito, D. Quaglia, J.C. De Martin, “Model-based Distortion Estimation for Perceptual Classification of Video Packets,” in the *Proceeding of IEEE Int. Workshop on Multimedia Signal Processing (MMSP)*, Siena, Italy, vol.1, pp.79-82, Sep. 2004.

[8] P. Bucciol, E. Masala, J.C. De Martin, “Perceptual ARQ for H.264 Video Streaming over 3G Wireless Networks”, *IEEE Int. Conf. On Communications (ICC)*, Paris, France, vol. 3, pp. 1288-1292, June 2004.

[9] P. Bucciol, E. Masala, E. Filippi, J.C. De Martin, "Cross-layer Perceptual ARQ for Video Communications over 802.11e Wireless Networks", *Hindawi Journal of Advances in Multimedia*, 2007.

[10] M. Baldi, J.C. De Martin, E. Masala, A. Vesco, “Distortion-Aware Video Communication with Pipeline Forwarding,” *ACM Multimedia 2006*, Santa Barbara, CA, USA, p. 117-120, Oct. 2006.

[11] A. Chou, Z. Miao, “Rate-Distorsion Opimized Streaming of Packetized Media,” *IEEE Transactions on multimedia*, vol.8, no.2, Apr. 2006.

[12] Gersho, Y. Shoham, “Efficient Bit Allocation for an Arbitrary Set of Quantizers”, *IEEE Transaction of Acoustics, Speech and Signal Processing*, vol.36, no.9, Sep. 1988.

[13] M. Coates, A.O. Hero, R.D. Nowak, B. Yu, “Internet Tomography,” *IEEE Signal Processing Magazine*, vol.19, pp. 47-65, May 2002.

[14] F. Lo Presti, N.G. Duffield, J. Horowitz, D. Towsley, “Multicast-based inference of network-internal delay distribution,” *IEEE Transaction Information Theory*, vol. 45, pp. 2462-2480, May 2002.

[15] M. Coates, R.D. Nowak, “Network loss inference using unicast end-to-end measurements,” in the *Proceeding of ITC Seminar IP Traffic, Measurements Modelling*, Monterey, CA, pp. 1-9, Sep. 2000.

[16] S. Martello, P. Toth, “Solution of the zero-one multiple knapsack problem,” *European Journal of Operational Research*, vol.4, pp.276-283, 1980.

[17] S. Martello, P. Toth, “A bound and bound algorithm for the zero-one multiple knapsack problem,” *Discrete Appl. Math*, vol.3, pp. 275-288, 1981.

[18] S. Martello, P. Toth, “Heuristic algorithms for the multiple knapsack problem,” *Computing 27*, pp. 93-112, 1981.

[19] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, “Joint Model v.11.0,” *URL: http://iphome.hhi.de/suehring/tml*, Jan. 2007.

[20] F. De Vito, L. Farinetti, J.C. De Martin, “Perceptual classification of MPEG video for Differentiated-Services communications,” in the *Proceeding IEEE Int. Conf. on Multimedia & Expo*, Lausanne, Switzerland, vol. 1, pp. 141-144, Aug. 2002.

[21] E. Masala, D. Quaglia, J.C. De Martin, “Adaptive picture slicing for distortion-based classification of video packets,” in the *Proceeding IEEE Workshop on Multimedia Signal Processing*, Cannes, France, pp. 111-116, Oct. 2001.

[22] R. Aravind, M.R. Civanlar, A.R. Reibman, “Packet loss resilience of MPEG-2 scalable video coding algorithms,” *IEEE Trans. on Circuits and Systems for Video Technology*, Vol 6, No 5, pp. 426-435, Oct. 1996.

[23] Rui Zhang, S.L. Regunathan, K. Rose, “End-to-end distortion estimation for RD-based robust delivery of pre-compressed video,” in the *Proceeding* of Asilomar Conference on Signals, Systems and Computers, vol.1, pp. 210-14, Nov. 2001.